On Two Characterizations of Feature Models

Two Characterizations of Feature Models

Ferruccio Damiani[1], Michael Lienhardt[2], and Luca Paolini[1]

[1] University of Turin, Italy
{ferruccio.damiani, luca.paolini}@unito.it
[2] ONERA, Palaiseau, France
michael.lienhardt@onera.fr

Abstract. Software-intensive systems can have thousands of interdependent configuration options across different subsystems. Feature models allow designers to organize the configuration space by describing configuration options using interdependent features: a feature is a name representing some functionality and each software variant is identified by set of features. Different representations of feature models have been proposed in the literature. In this paper we focus on the propositional representation (which works well in practice) and the extensional representation (which has been recently shown well suited for theoretical investigations). We provide an algebraic and a propositional characterization of novel and existing feature model operations and relations, and we formalize the connection between the two characterizations as monomorphisms from lattices of propositional feature models to lattices of extensional features models. These results shed new light on the correspondence between the extensional and the propositional representations of feature models. Thus paving the way towards facilitating practical exploitation, by relying on the propositional representation, of theoretical developments expressed by relying on the extensional representation.

1 Introduction

Software-intensive systems can have thousands of interdependent configuration options across different subsystems. In the resulting configuration space, different software variants can be obtained by selecting among these configuration options and accordingly assembling the underlying subsystems. The interdependencies between options are dictated by corresponding interdependencies between the underlying subsysems [5].

Feature models [6] allow developers to organize the configuration space and facilitate the construction of software variants by describing configuration options using interdependent *features* [18]: a feature is a name representing some functionality, a set of features is called a *configuration*, and each configuration that fulfills the interdependencies expressed by the feature model (called a *product*) identifies a software variant.

Software-intensive systems can comprise thousands of features and several subsystems [10, 9, 27, 19]. The design, development and maintenance of feature models with thousands of features can be simplified by representing large feature models as sets of smaller interdependent feature models [9, 24] which we call *fragments*. To this aim, several representations of feature models have been proposed in the literature (see, e.g., [6] and Sect. 2.3 of Apel *et al.* [4]) and many approaches for composing feature models from fragments have been investigated [1, 3, 12, 13, 23, 26].

In this paper we focus on the propositional representation (which works well in practice [22, 8, 28]) and the extensional representation (which has been recently shown well suited for theoretical investigations [25, 20]). We investigate the correspondence between these two representation and between the corresponding formulation of existing and novel feature model operators and relations. The starting point of this investigation is a novel partial order between feature models (the feature model fragment relation), which is induced by a notion of feature model composition that has been used to model industrial-size configuration spaces [25, 20]. We exploit this partial order to provide an algebraic characterization of feature model operations and relations, we provide a propositional characterizations of them, and we formalize the connection between the two characterizations as monomorphisms from lattices of propositional feature models to lattices of extensional features models.

The remainder of this paper is organized as follows. In Section 2 we recollect the necessary background and introduce the feature model fragment relation. In Section 3 we present the algebraic characterization of feature model operations and relations, and in Section 4 we present the propositional characterization of the operations and relations together with a formal account of the relation between the two characterizations. We discuss related related work in Section 5, and conclude the paper in Section 6 by outlining planned future work.

2 Background and Concept

This section presents a formalization of feature models (FM) and related notions, including feature model interfaces and composition.

2.1 Feature Model Representations and Analyses

In this paper, we focus on the propositional and on the extensional representations of feature models (see, e.g., Batory [6] and Sect. 2.3 of Apel *et al.* [4] for a discussion about other representations).

Definition 1 (Feature model, propositional representation). *A propositional feature model Φ is a pair $(\mathcal{F}, \phi)$ where $\mathcal{F}$ is a set of features and ϕ is a propositional formula whose variables x are elements of $\mathcal{F}$:*

$$\phi = x \;\mid\; \phi \wedge \phi \;\mid\; \phi \vee \phi \;\mid\; \phi \rightarrow \phi \;\mid\; \neg\phi \;\mid\; \mathsf{false} \;\mid\; \mathsf{true}$$

Its products are the set of features $p \subseteq \mathcal{F}$ such that ϕ is satisfied by assigning value true *to the variables x in p and* false *to the variables in $\mathcal{F} \setminus p$.*

The ICSE2020 encoding of the gentoo package made sense for that paper, but here it doesn't allow to show things like conflicts creating dead features and such. Hence in this paper, we should consider that the feature model encoding only consider the configuration of the package, not the possibility of it not being installed. This means that all of the example in this section must be changed.

Example 1 (A propositional representation of the glibc feature model). Gentoo packages can be configured by selecting features (called *use flags* in Gentoo), which may trigger dependencies or conflicts between packages. Version 2.29 of the *glibc* library, that contains the core functionalities of most Linux systems, is provided by the package sys-libs/glibc-2.29-r2 (abbreviated to glibc in the sequel). This package has many dependencies, including (as expressed in Gentoo's notation):

```
doc? ( sys−apps/texinfo )
    vanilla?( !sys−libs/timezone−data )
```

This dependency expresses that glibc requires the texinfo documentation generator (provided by any version of the sys-apps/texinfo package) whenever the feature doc is selected and if the feature vanilla is selected, then glibc conflicts with any version of the time zone database (as stated with the !sys-libs/timezone-data constraint). These *dependencies* can be expressed by a feature model $(\mathcal{F}_{\mathsf{glibc}}, \phi_{\mathsf{glibc}})$ where

cannot talk about conflict here, since it is a key notion in the rest of the paper

$$\mathcal{F}_{\mathsf{glibc}} = \{\mathsf{glibc},\ \mathsf{txinfo},\ \mathsf{tzdata},\ \mathsf{glibc{:}doc},\ \mathsf{glibc{:}v}\}$$
$$\phi_{\mathsf{glibc}} = \mathsf{glibc} \wedge (\mathsf{glibc{:}doc} \rightarrow \mathsf{txinfo}) \wedge (\mathsf{glibc{:}v} \rightarrow (\neg \mathsf{tzdata}))$$

Here, the feature glibc represents the glibc package; txinfo represents any sys-apps/texinfo package; tzdata represents any version of the sys-libs/timezone-data package; and glibc:doc (resp. glibc:v) represents the glibc's doc (resp. vanilla) use flag.

The propositional representation of feature models works well in practice [22, 8, 28]. Recently, Schröter *et al.* [25] pointed out that using an extensional representation of feature models simplifies the presentation of feature model concepts and proofs.

Definition 2 (Feature model, extensional representation). *An* extensional feature model $\mathcal{M}$ *is a pair* $(\mathcal{F}, \mathcal{P})$ *where* $\mathcal{F}$ *is a set of features and* $\mathcal{P} \subseteq 2^{\mathcal{F}}$ *a set of products.*

Example 2 (An extensional representation of the glibc feature model). Let 2^S denote the powerset of S. The feature model of Example 1 can be given an extensional representation $\mathcal{M}_{\mathsf{glibc}} = (\mathcal{F}_{\mathsf{glibc}}, \mathcal{P}_{\mathsf{glibc}})$ where $\mathcal{F}_{\mathsf{glibc}}$ is the same as in Example 1 and

Informally, the above definition can be also found in [4]; quoting from p.27: "In its simplest form, a feature model comprises a list of features and an enumeration of all valid feature combinations." Indeed, the book just consider finite set of features, because this is the only relevant case in practice.

$$\begin{aligned}\mathcal{P}_{\mathsf{glibc}} = &\{\{\mathsf{glibc}\}, \{\mathsf{glibc},\ \mathsf{txinfo}\}, \{\mathsf{glibc},\ \mathsf{tzdata}\}, \{\mathsf{glibc},\ \mathsf{txinfo},\ \mathsf{tzdata}\}\} \cup \\ &\{\{\mathsf{glibc},\ \mathsf{glibc{:}doc},\ \mathsf{txinfo}\}, \{\mathsf{glibc},\ \mathsf{glibc{:}doc},\ \mathsf{txinfo},\ \mathsf{tz\text{-}data}\}\} \cup \\ &\{\{\mathsf{glibc},\ \mathsf{glibc{:}v}\}, \{\mathsf{glibc},\ \mathsf{glibc{:}v},\ \mathsf{txinfo}\}\} \cup \\ &\{\{\mathsf{glibc},\ \mathsf{glibc{:}doc},\ \mathsf{glibc{:}v},\ \mathsf{txinfo}\}\}\end{aligned}$$

In the description of $\mathcal{P}_{\mathsf{glibc}}$, the first line contains products with glibc but none of its use flags are selected, so texinfo and tz-data can be freely installed; the second line contains products with the use flag doc selected in glibc, so a package of sys-apps/texinfo is always required; the third line contains products with the use flag vanilla selected in glibc, so no package of sys-libs/timezone-data is allowed; finally, the fourth line contains products with both glibc's use flags selected, so sys-apps/texinfo is mandatory and sys-libs/timezone-data forbidden.

Definition 3 (Empty feature model and void feature models). *The* empty *feature model, denoted $\mathcal{M}_\emptyset = (\emptyset, \{\emptyset\})$, has no features and has just the empty product $\emptyset$. A* void *feature model is a feature model that has no products, i.e., of the form $\mathcal{M}^{\mathcal{F}} = (\mathcal{F}, \emptyset)$ for some set of features $\mathcal{F}$.*

2.2 Feature Model Composition

Complex software systems, like the Gentoo source-based Linux distribution [17], often consist of many interdependent configurable packages [21, 19, 20]. The configuration options of each package can be represented by a feature model. Therefore, configuring two packages in such a way that they can be installed together corresponds to finding a product in the composition of their associated feature models. ~~In~~ We prove that in the propositional representation of feature models this composition corresponds to logical conjunction: the composition of two feature models $(\mathcal{F}_1, \phi_1)$ and $(\mathcal{F}_2, \phi_2)$ is the feature model

$$(\mathcal{F}_1 \cup \mathcal{F}_2, \phi_1 \wedge \phi_2).$$

In the extensional representation of feature models, this composition corresponds to the binary operator $\bullet$ of Schröter *et al.* [25]. This operator, which is similar to the join operator from relational algebra [14], yields all combinations between both product sets.

Definition 4 (Feature model composition). *The composition of two feature models $\mathcal{M}_1 = (\mathcal{F}_1, \mathcal{P}_1)$ and $\mathcal{M}_2 = (\mathcal{F}_2, \mathcal{P}_2)$, denoted $\mathcal{M}_1 \bullet \mathcal{M}_2$, is the feature model defined by:*

$$\mathcal{M}_1 \bullet \mathcal{M}_2 = (\mathcal{F}_1 \cup \mathcal{F}_2, \{p \cup q \mid p \in \mathcal{P}_1, q \in \mathcal{P}_2, p \cap \mathcal{F}_2 = q \cap \mathcal{F}_1\}).$$

The composition operator $\bullet$ is associative and commutative, with $\mathcal{M}_\emptyset$ as identity element (i.e., $\mathcal{M} \bullet \mathcal{M}_\emptyset = \mathcal{M}$). Composing a feature model with a void feature model yields a void feature model: $(\mathcal{F}_1, \mathcal{P}_1) \bullet (\mathcal{F}_2, \emptyset) = (\mathcal{F}_1 \cup \mathcal{F}_2, \emptyset)$.

Example 3 (Composing glibc and gnome-shell feature models). Let us consider another important package of the Gentoo distribution: *gnome-shell*, a core component of the Gnome Desktop environment. Version 3.30.2 of *gnome-shell* is provided by the package gnome-base/gnome-shell-3.30.2-r2 (abbreviated to g-shell in the sequel), and its dependencies include the following statement:

```
networkmanager?( sys-libs/timezone-data )
```

This dependency expresses that g-shell requires any version of the time zone database when the feature networkmanager is selected.

The *propositional representation* of this dependency can be captured by the feature model $(\mathcal{F}_{\textsf{g-shell}}, \phi_{\textsf{g-shell}})$, where

$$\mathcal{F}_{\textsf{g-shell}} = \{\textsf{g-shell, tzdata, g-shell:nm}\}$$
$$\phi_{\textsf{g-shell}} = \textsf{g-shell} \wedge (\textsf{g-shell:nm} \rightarrow \textsf{tzdata})$$

The corresponding *extensional representation* of this feature model is $\mathcal{M}_{\textsf{g-shell}} = (\mathcal{F}_{\textsf{g-shell}}, \mathcal{P}_{\textsf{g-shell}})$, where:

$$\begin{aligned}\mathcal{P}_{\textsf{g-shell}} =& \{\{\textsf{g-shell}\}, \{\textsf{g-shell, tzdata}\}\cup \\ & \{\{\textsf{g-shell, tzdata, g-shell:nm}\}\}\end{aligned}$$

Here, the first line contains products with g-shell but none of its use flags are selected: tzdata can be freely selected; the second line is the product where g-shell:nm is also selected and tzdata becomes mandatory; finally, the third line represents products without g-shell.

The *propositional representation* of the composition is the feature model $(\mathcal{F}_{\textsf{full}}, \phi_{\textsf{full}})$, where

$$\begin{aligned}\mathcal{F}_{\textsf{full}} &= \mathcal{F}_{\textsf{glibc}} \cup \mathcal{F}_{\textsf{g-shell}} \\ &= \{\textsf{glibc, txinfo, tzdata, g-shell, glibc:doc, glibc:v, g-shell:nm}\} \\ \phi_{\textsf{full}} &= \phi_{\textsf{glibc}} \wedge \phi_{\textsf{g-shell}} \\ &= (\textsf{glibc} \wedge ((\textsf{glibc:doc} \rightarrow \textsf{txinfo}) \wedge (\textsf{glibc:v} \rightarrow (\neg \textsf{tz-data})))\wedge \\ &\quad (\textsf{g-shell} \wedge (\textsf{g-shell:nm} \rightarrow \textsf{tzdata}))\end{aligned}$$

The *extensional representation* of the composition is the feature model $\mathcal{M}_{\textsf{full}} = \mathcal{M}_{\textsf{glibc}} \bullet \mathcal{M}_{\textsf{g-shell}} = (\mathcal{F}_{\textsf{full}}, \mathcal{P}_{\textsf{full}})$ where

$$\begin{aligned}\mathcal{P}_{\textsf{full}} = & \{\{\textsf{glibc, g-shell}\} \cup p \mid p \in 2^{\{\textsf{txinfo, tzdata}\}}\} \cup \\ & \{\{\textsf{glibc, glibc:doc, txinfo, g-shell}\} \cup p \mid p \in 2^{\{\textsf{tzdata}\}}\} \cup \\ & \{\{\textsf{glibc, glibc:v, g-shell}\} \cup p \mid p \in 2^{\{\textsf{txinfo}\}}\} \cup \\ & \{\{\textsf{glibc, g-shell, g-shell:nm, tzdata}\} \cup p \mid p \in 2^{\{\textsf{txinfo}\}}\} \cup \\ & \{\{\textsf{glibc, glibc:doc, glibc:v, txinfo, g-shell}\}\} \cup \\ & \{\{\textsf{glibc, glibc:doc, txinfo, g-shell, g-shell:nm, tzdata}\}\}\end{aligned}$$

Here, the first line contains the products where both glibc and g-shell are installed, but without use flags selected, so all optional package can be freely selected; the second line contains the products with the glibc's use flag doc selected, so sys-apps/texinfo becomes mandatory; the third line contains the products with the glibc's use flag vanilla selected, so sys-libs/timezone-data is forbidden; the fourth line contains the products with the g-shell's use flag vanilla network manager, so sys-libs/timezone-data is mandatory; the fifth line contains the product with glibc's both use flags selected and the sixth line contains the product with glibc's use flag doc and g-shell's use flag networkmanager are selected.

2.3 Feature Model Slices and Interfaces

Io postporrei questa subsection alla successiva. Anticipare la successiva gli da piu enfasi e mi sembra semplifichi la narrazione

Feature model slices were defined by Acher et al. [2] as a unary operator Π_Y that that restricts a feature model to the set Y of features. In order to simplify the presentation of the slice operator, we introduce the following auxiliary notation (where $\mathcal{P}$ is a set of products and Y is a set of features): $\mathcal{P}|_Y = \{p \cap Y \mid p \in \mathcal{P}\}$.

Definition 5 (Feature model slice operator). *Let $\mathcal{M} = (\mathcal{F}, \mathcal{P})$ be a feature model. The* slice *operator Π_Y on feature models, where Y is a set of features, is defined by: $\Pi_Y(\mathcal{M}) = (\mathcal{F} \cap Y, \mathcal{P}|_Y)$.*

More recently, Schröter et al. [25] introduced the following notion of feature model interface.

Definition 6 (Feature model interface relation). *A feature model $\mathcal{M}_0 = (\mathcal{F}_0, \mathcal{P}_0)$ is an* interface *of feature model $\mathcal{M} = (\mathcal{F}, \mathcal{P})$, denoted as $\mathcal{M}_0 \preceq \mathcal{M}$, whenever both $\mathcal{F}_0 \subseteq \mathcal{F}$ and $\mathcal{P}_0 = \mathcal{P}|_{\mathcal{F}_0}$ hold.*

Remark 1 (Feature model interfaces and slices are equivalent). As pointed out in [25], feature model slices and interfaces are closely related. Namely: $\mathcal{M}_0 \preceq \mathcal{M}$ holds if and only if there exists a set of features Y such that $\mathcal{M}_0 = \Pi_Y(\mathcal{M})$.

Example 4 (A slice of the glibc feature model). Applying the operator $\Pi_{\{\mathsf{glibc},\ \mathsf{glibc:v}\}}$ to the feature model $\mathcal{M}_{\mathsf{glibc}}$ of Example 2 yields the feature model

$$\mathcal{F} = \{\mathsf{glibc},\ \mathsf{glibc:v}\}$$
$$\mathcal{P} = \{\emptyset, \{\mathsf{glibc}\}, \{\mathsf{glibc},\ \mathsf{glibc:v}\}\},$$

which (according to Remark 1) is is an interface for $\mathcal{M}_{\mathsf{glibc}}$.

2.4 Feature Model Components

The notion of feature model composition induces the following notion of feature model component.

Definition 7 (Feature model component relation). *A feature model $\mathcal{M}_0$ is a* component *of a feature model $\mathcal{M}$, denoted as $\mathcal{M}_0 \leq \mathcal{M}$, whenever there exists a feature model $\mathcal{M}_1$ such that $\mathcal{M}_0 \bullet \mathcal{M}_1 = \mathcal{M}$.*

In practice, this relation captures at the level of feature models the fact that the combination of two configurable programs contains both programs: the composition of the packages glibc and g-shell contains both packages, and its feature model consequently contains the feature models of both packages. We have (by definition) that $\mathcal{M}_{\mathsf{g\text{-}shell}} \leq (\mathcal{M}_{\mathsf{g\text{-}shell}} \bullet \mathcal{M}_{\mathsf{glibc}})$.

Interestingly, this relation has some counter-intuitive properties: ~~some configurations might not be possible anymore.~~ if $\mathcal{M}_0 \leq \mathcal{M}$ then some configurations of $\mathcal{M}_0$ might not be possible anymore in $\mathcal{M}$.

Example 5. Consider for instance the version 3.0.8 of the library libical in Gentoo. Its feature model contains the following constraint (as expressed in Gentoo notation):

```
berkdb? ( sys-libs/db ) sys-libs/timezone-data
```

This dependency expresses that libical requires the db library whenever the feature berkdb is selected and requires the package sys-libs/timezone-data to be installed. These *dependencies* can be extensionally expressed by a feature model $\mathcal{M}_{\mathsf{libical}}(\mathcal{F}_{\mathsf{libical}}, \mathcal{P}_{\mathsf{libical}})$ where

$$\begin{aligned}\mathcal{F}_{\mathsf{libical}} &= \{\mathsf{libical},\ \mathsf{berkdb},\ \mathsf{sys\text{-}libs/db},\ \mathsf{tzdata}\}\\ \mathcal{P}_{\mathsf{libical}} &= \{\{\mathsf{libical},\ \mathsf{tzdata}\}, \{\mathsf{libical},\ \mathsf{tzdata},\ \mathsf{berkdb},\ \mathsf{sys\text{-}libs/db}\}\}\end{aligned}$$

Composing the feature model of glibc and libical raises the feature model $\mathcal{M}_c = (\mathcal{F}_c, \mathcal{P}_c)$ where $\mathcal{F}_c = \mathcal{F}_{\mathsf{glibc}} \cup \mathcal{F}_{\mathsf{libical}}$ and

of the composition is the feature model $\mathcal{M}_{\mathsf{full}} = \mathcal{M}_{\mathsf{glibc}} \bullet \mathcal{M}_{\mathsf{g\text{-}shell}} = (\mathcal{F}_{\mathsf{full}}, \mathcal{P}_{\mathsf{full}})$ where

@MICHAEL': something is missing here.

$$\begin{aligned}\mathcal{P}_c = \{&\{\mathsf{glibc},\ \mathsf{libical},\ \mathsf{tzdata}\} \cup p \mid p \in 2^{\{\mathsf{txinfo},\ \mathsf{sys\text{-}libs/db}\}}\} \cup\\ &\{\{\mathsf{glibc},\ \mathsf{glibc{:}doc},\ \mathsf{txinfo},\ \mathsf{libical},\ \mathsf{tzdata}\} \cup p \mid p \in 2^{\{\mathsf{sys\text{-}libs/db}\}}\} \cup\\ &\{\{\mathsf{glibc},\ \mathsf{libical},\ \mathsf{berkdb},\ \mathsf{sys\text{-}libs/db},\ \mathsf{tzdata}\} \cup p \mid p \in 2^{\{\mathsf{txinfo}\}}\} \cup\\ &\{\{\mathsf{glibc},\ \mathsf{glibc{:}doc},\ \mathsf{txinfo},\ \mathsf{libical},\ \mathsf{berkdb},\ \mathsf{sys\text{-}libs/db},\ \mathsf{tzdata}\}\}\end{aligned}$$

Here, the first line contains the products where both glibc and libical are installed, but without use flags selected, so only the annex package timezone-data is mandatory; the second line contains the products with the glibc's use flag doc selected, so sys-apps/texinfo becomes mandatory; the third line contains theproducts with the libical's use flag berkdb, so sys-libs/db becomes mandatory; finally, the fourth line contains the product with all optional features of both glibc and libical selected.

It is easy to see from the constraint, and also from the extensional representation, that combining glibc and libical makes the feature glibc:v dead. When composed, the feature models interract and not all combination of product are available.

This property is counter-intuitive, since combining two programs make the declarations of both programs available, but combining two configurable programs is far more subtle, due to interractions between the two configuration spaces.

In the following section, we analyse that order relation and identify precisely its counter-intuitive property. We will then use this to construct the rest of the paper.

@MICHAEL: please complete, to match the abstract. Consider changing accordingly the examples in the previous subsections.

====================

@MICHAEL: la relazione $\leq$ andrebbe motivata in termini di gentoo. Se non bastasse, magari si potrebbe parlare (anche) di Elsa.

3 Algebraic Characterization of Feature Models

In Section 3.1 we recall some relevant algebraic notions. In Section 3.2 we show that the feature model component relation induces a lattice of feature models where the join operation is feature model composition. Then, in Section 3.3 we show that the feature model component relation generalizes the feature model interface relation and provide some algebraic properties of the feature model slice operation.

3.1 A Recollection of Algebraic Notions

In this section we briefly recall the notions of lattice, bounded lattice and Boolean algebra (see, e.g., Davey and Priestley [16] for a detailed presentation). An *ordered lattice* is a partially ordered set $(P, \sqsubseteq)$ such that, for every $x, y \in P$, both the least upper bound (lub) of $\{x, y\}$, denoted $\inf\{x, y\} = \min\{a \mid x, y \leq a\}$, and the greatest lower bound (glb) of $\{x, y\}$, denoted $\inf\{x, y\} = \max\{a \mid a \leq x, y\}$, are always defined.

An *algebraic lattice* is an algebraic structure $(L, \sqcup, \sqcap)$ where L is non-empty set equipped with two binary operations $\sqcup$ (called *join*) and $\sqcap$ (called *meet*) which satisfy the following:

- Associative laws: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$, $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$.
- Commutative laws: $x \sqcup y = y \sqcup x$, $x \sqcap y = y \sqcap x$.
- Absorption laws: $x \sqcup (x \sqcap y) = x$, $x \sqcap (x \sqcup y) = x$.
- Idempotency laws: $x \sqcup x = x$, $x \sqcup x = x$.

As known, the two notions of lattice are equivalent (Theorem 2.9 and 2.10 of [16]). In particular, given an ordered lattice $(P, \sqsubseteq)$ with the operations $x \sqcup y = \sup\{x, y\}$ and $x \sqcap y = \inf\{x, y\}$, the following three statements are equivalent (Theorem 2.8 of [16]):

- $x \sqsubseteq y$,
- $x \sqcup y = y$,
- $x \sqcap y = x$.

A *bounded lattice* is a lattice that contains two elements $\bot$ (the lattice's *bottom*) and $\top$ (the lattice's *top*) which satisfy the following: $\bot \sqsubseteq x \sqsubseteq \top$. Let L be a bounded lattice, $y \in L$ is a *complement* of $x \in L$ if $x \sqcap y = \bot$ and $x \sqcup y = \top$. If x has a unique complement, we denote this complement by $\bar{x}$.

A *distributive lattice* is a lattice which satisfies the following distributive law: $x \sqcap (y \sqcup z) = (x \sqcup y) \sqcap (x \sqcup z)$. In a bounded distributive lattice the complement (whenever it exists) is unique (see [16, Section 4.13]).

A *Boolean lattice* (a.k.a. *Boolean algebra*) L is a bounded distributive lattice such that each $x \in L$ has a (necessarily unique) complement $\bar{x} \in L$.

3.2 Lattices of Feature Models

Although only finite feature models are relevant in practice, in our theoretical development (in order to enable a better understanding of the relation between the extensional and the propositional representations) we consider also feature models with infinitely many features and products. The following definition introduces a notation for three different sets of extensional feature models (see Definition 2) over a given set of features.

Definition 8 (Sets of extensional feature models over a set of features). *Let X be a set of features. We denote:*

- $\mathfrak{E}(X)$ *the set of the extensional feature models* $(\mathcal{F},\mathcal{P})$ *such that* $\mathcal{F} \subseteq X$*;*
- $\mathfrak{E}_{\mathrm{fin}}(X)$ *the subset of the finite elements of* $\mathfrak{E}(X)$*, i.e.,* $(\mathcal{F},\mathcal{P})$ *such that* $\mathcal{F} \subseteq_{\mathrm{fin}} X$*; and*
- $\mathfrak{E}_{\mathrm{eql}}(X)$ *the subset of elements of* $\mathfrak{E}(X)$ *that have exactly the features* X*, i.e.,* $(\mathcal{F},\mathcal{P})$ *such that* $\mathcal{F} = X$*.*

Note that, if X has infinitely many elements then $\mathfrak{E}_{\mathrm{fin}}(X)$ has infinitely many elements too. Instead, if X is finite then $\mathfrak{E}(X)$ and $\mathfrak{E}_{\mathrm{fin}}(X)$ ~~are the same~~ coincide and have a finite number of elements. For this reason, in the following, we consider $\mathfrak{E}_{\mathrm{fin}}(X)$ only under the hypothesis that X is infinite.

Lemma 1 (Two criteria for the feature model component relation). *Given a set X, for all $\mathcal{M}_1 = (\mathcal{F}_1,\mathcal{P}_1)$ and $\mathcal{M}_2 = (\mathcal{F}_2,\mathcal{P}_2)$ in $\mathfrak{E}(X)$, the following properties are equivalent:*

i) $\mathcal{M}_1 \leq \mathcal{M}_2$
ii) $\mathcal{M}_1 \bullet \mathcal{M}_2 = \mathcal{M}_2$
iii) $\mathcal{F}_1 \subseteq \mathcal{F}_2$ *and* $\mathcal{P}_1 \supseteq \mathcal{P}_2|_{\mathcal{F}_1}$

Proof. *i)* $\Rightarrow$ *ii)***.** It is easy to check that $\mathcal{M}' \bullet \mathcal{M}' = \mathcal{M}'$, for all $\mathcal{M}'$. We have

$$\begin{aligned}
\mathcal{M}_1 \bullet \mathcal{M}_2 &= \mathcal{M}_1 \bullet (\mathcal{M}_1 \bullet \mathcal{M}) && \text{for some } \mathcal{M} \in \mathfrak{E}(X)\\
&= (\mathcal{M}_1 \bullet \mathcal{M}_1) \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X)\\
&= \mathcal{M}_1 \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X)\\
&= \mathcal{M}_2
\end{aligned}$$

ii) $\Rightarrow$ *iii)***.** By definition of $\bullet$, it is clear from the hypothesis that $\mathcal{F}_1 \subseteq \mathcal{F}_2$. Moreover, if we write $S = \mathcal{P}_2|_{\mathcal{F}_1}$, the hypothesis say us that $\mathcal{P}_2 = \{q \mid q \in \mathcal{P}_2, q \cap \mathcal{F}_1 \in (\mathcal{P}_1 \cap S)\}$. We can thus conclude with the following equivalences:

$$\begin{aligned}
\mathcal{P}_2 = \{q \mid q \in \mathcal{P}_2, q \cap \mathcal{F}_1 \in (\mathcal{P}_1 \cap S)\}\\
&\Leftrightarrow \forall q \in \mathcal{P}_2,\ q \cap \mathcal{F}_1 \in (\mathcal{P}_1 \cap S)\\
&\Leftrightarrow \{q \cap \mathcal{F}_1 \mid q \in \mathcal{P}_2\} \subseteq (\mathcal{P}_1 \cap S)\\
&\Leftrightarrow S \subseteq (\mathcal{P}_1 \cap S)\\
&\Leftrightarrow S \subseteq \mathcal{P}_1
\end{aligned}$$

Moreover, $\mathcal{P}_2 = \{p \cup q \mid p \in \mathcal{P}_1, q \in \mathcal{P}_2, p \cap \mathcal{F}_2 = q \cap \mathcal{F}_1\}$ immediately implies that $\mathcal{P}_2 = \{q \mid p \in \mathcal{P}_1, q \in \mathcal{P}_2, p = q \cap \mathcal{F}_1\}$, which in turn implies $\mathcal{P}_2|_{\mathcal{F}_1} \subseteq \mathcal{P}_1$.

iii) $\Rightarrow$ *i)*. Let still write $S = \mathcal{P}_2|_{\mathcal{F}_1}$. We have:

$$\begin{aligned}
\mathcal{M}_1 \bullet \mathcal{M}_2 &= (\mathcal{F}_1 \cup \mathcal{F}_2, \{p \cup q \mid p \in \mathcal{P}_1,\ q \in \mathcal{P}_2,\ p \cap \mathcal{F}_2 = q \cap \mathcal{F}_1\}) \\
&= (\mathcal{F}_2, \{q \mid q \in \mathcal{P}_2, q \cap \mathcal{F}_1 \in \mathcal{P}_1\}) \\
&= (\mathcal{F}_2, \{q \mid q \in \mathcal{P}_2, q \cap \mathcal{F}_1 \in (\mathcal{P}_1 \cap S)\} \\
&\quad \cup \{q \mid q \in \mathcal{P}_2,\ q \cap \mathcal{F}_1 \in \{p \mid p \in \mathcal{P}_1, \forall q \in \mathcal{P}_2, p \neq q \cap \mathcal{F}_1\}\}) \\
&= (\mathcal{F}_2, \{q \mid q \in \mathcal{P}_2, q \cap \mathcal{F}_1 \in (\mathcal{P}_1 \cap S)\}) \\
&= (\mathcal{F}_2, \{q \mid q \in \mathcal{P}_2, q \cap \mathcal{F}_1 \in S)\}) \\
&= \mathcal{M}_2
\end{aligned}$$

By hypothesis, $(\mathcal{F}_1 \cup \mathcal{F}_2, \{p \cup q \mid p \in \mathcal{P}_1,\ q \in \mathcal{P}_2,\ p \cap \mathcal{F}_2 = q \cap \mathcal{F}_1\}) = (\mathcal{F}_2, \mathcal{P}_2)$, thus $\mathcal{M}_1 \bullet \mathcal{M}_2 = \mathcal{M}_2$. This implies, by definition of $\leq$, that $\mathcal{M}_1 \leq \mathcal{M}_2$. $\square$

Theorem 1 (Lattices of feature models over a set of features). *Given a set X and two feature models $\mathcal{M}_1 = (\mathcal{F}_1, \mathcal{P}_1), \mathcal{M}_2 = (\mathcal{F}_2, \mathcal{P}_2) \in \mathfrak{E}(X)$, we define: $\mathcal{M}_1 \star \mathcal{M}_2 = (\mathcal{F}_1 \cap \mathcal{F}_2,\ \mathcal{P}_1|_{\mathcal{F}_2} \cup \mathcal{P}_2|_{\mathcal{F}_1})$, and $\overline{\mathcal{M}_1} = (\mathcal{F}_1, 2^{\mathcal{F}_1} \setminus \mathcal{P}_1)$. Then:*

1. *$(\mathfrak{E}(X), \leq)$ is a bounded lattice with join $\bullet$, meet $\star$, bottom $\mathcal{M}_\emptyset = (\emptyset, \{\emptyset\})$ and top $(X, \emptyset)$.*
2. *If X ~~has infinitely many elements,~~ is an infinite set then $\mathfrak{E}_{\text{fin}}(X)$ is a sublattice of $\mathfrak{E}(X)$ with the same bottom and no top.*
3. *$\mathfrak{E}_{\text{eql}}(X)$ is a sublattice of $\mathfrak{E}(X)$ which is a Boolean lattice with bottom $(X, 2^X)$, same top of $\mathfrak{E}(X)$, and complement $^-$.*

Proof. **Part 1.1: $\leq$ is a partial order on $\mathfrak{E}(X)$.** For any $\mathcal{M}_1 \leq \mathcal{M}_2 \leq \mathcal{M}_3 \in \mathfrak{E}(X)$, we have

- Reflexivity: $\mathcal{M}_1 \bullet \mathcal{M}_\emptyset = \mathcal{M}_1$
- Antisymmetry: suppose additionally $\mathcal{M}_2 \leq \mathcal{M}_1$. We have

$$\begin{aligned}
\mathcal{M}_1 &= \mathcal{M}_2 \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X) \\
&= \mathcal{M}_1 \bullet \mathcal{M}' \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X) \\
&= \mathcal{M}_1 \bullet \mathcal{M}' \bullet \mathcal{M}' \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X) \\
&= \mathcal{M}_2 \bullet \mathcal{M}' \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X) \\
&= \mathcal{M}_2 \bullet \mathcal{M} \bullet \mathcal{M}' && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X) \\
&= \mathcal{M}_1 \bullet \mathcal{M}' && \text{for some } \mathcal{M}' \in \mathfrak{E}(X) \\
&= \mathcal{M}_2
\end{aligned}$$

- Transitivity:

$$\begin{aligned}
\mathcal{M}_3 &= \mathcal{M}_1 \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X) \\
&= (\mathcal{M}_1 \bullet \mathcal{M}') \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X) \\
&= \mathcal{M}_1 \bullet (\mathcal{M}' \bullet \mathcal{M}) && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X)
\end{aligned}$$

@MICHAEL: qui le cose sono state copiate dall'appendice in modo approssimativo senza neanche leggere i commenti che avevo lasciato ... io ho messo delle pezze, ma non ho molto tempo: per favore cura di piu il tuo lavoro

Part 1.2: $(\mathfrak{E}(X), \leq)$ is a lattice with $\mathcal{M}_\emptyset$ as bottom and $(X, \emptyset)$ as top. Let $\uparrow \mathcal{M}$ be the set of upper bounds of $\mathcal{M}$ w.r.t. $\leq$, viz. $\{\mathcal{M}' \mid \mathcal{M} \leq \mathcal{M}'\}$; and, let $\downarrow \mathcal{M}$ be the set of lower bounds of $\mathcal{M}$ w.r.t. $\leq$, viz. $\{\mathcal{M}' \mid \mathcal{M}' \leq \mathcal{M}\}$.

– By definition of $\leq$, we have $\mathcal{M}_1 \bullet \mathcal{M}_2 \in (\uparrow \mathcal{M}_1) \cap (\uparrow \mathcal{M}_2)$. Moreover, ~~with $\mathcal{M} \in (\uparrow \mathcal{M}_x) \cap (\uparrow \mathcal{M}_y)$,~~ for all common upper bound $\mathcal{M} \in (\uparrow \mathcal{M}_1) \cap (\uparrow \mathcal{M}_2)$, we have (cf. Lemma 1)

$$\mathcal{M} = \mathcal{M}_1 \bullet \mathcal{M} = \mathcal{M}_1 \bullet (\mathcal{M}_2 \bullet \mathcal{M}) = (\mathcal{M}_1 \bullet \mathcal{M}_2) \bullet \mathcal{M}$$

And so we have ~~$\mathcal{M}_1 \sqcap \mathcal{M}_2 = \mathcal{M}_1 \bullet \mathcal{M}_2$.~~ $\mathcal{M}_1 \sqcup \mathcal{M}_2 = \mathcal{M}_1 \bullet \mathcal{M}_2$.

– Let $\mathcal{M} = (\mathcal{F}, \mathcal{P}) = (\mathcal{F}_1 \cap \mathcal{F}_2, \mathcal{P}_1 |_{\mathcal{F}_2} \cup \mathcal{P}_2 |_{\mathcal{F}_1})$. We have $\{p \cap \mathcal{F} \mid p \in \mathcal{P}_i\} \subseteq \mathcal{P}$ and $\mathcal{F} \subseteq \mathcal{F}_i$ for $i \in \{1, 2\}$: we thus have ~~$\mathcal{M} \in \inf(\mathcal{M}_1) \cap \inf(\mathcal{M}_2)$.~~ $\mathcal{M} \in (\downarrow \mathcal{M}_1) \cap (\downarrow \mathcal{M}_2)$. Moreover, ~~with $(\mathcal{F}', \mathcal{P}') \in \inf(\mathcal{M}_1) \cap \inf(\mathcal{M}_2)$, we have that $\mathcal{F}' \subseteq \mathcal{F}_i$ and $\{p \cap \mathcal{F}' \mid p \in \mathcal{P}_i\} \subseteq \mathcal{P}'$ for $i \in \{1, 2\}$. Hence, we have $\mathcal{F}' \subseteq \mathcal{F}$ and $\{p \cap \mathcal{F}' \mid p \in \mathcal{P}\} \subseteq \mathcal{P}'$, which implies that $(\mathcal{F}', \mathcal{P}') \leq \mathcal{M}$.~~ for all $(\mathcal{F}', \mathcal{P}') \in (\downarrow \mathcal{M}_1) \cap (\downarrow \mathcal{M}_2)$, it is easy to see that, $\mathcal{F}' \subseteq \mathcal{F}_1 \cap \mathcal{F}_2 \subseteq \mathcal{F}$ and $\mathcal{P}|_{\mathcal{F}'} \subseteq \mathcal{P}'$ by Lemma 1. And so we have ~~$\mathcal{M}_1 \sqcup \mathcal{M}_2 = \mathcal{M}$.~~ $\mathcal{M}_1 \sqcap \mathcal{M}_2 = \mathcal{M}$.

As for all $\mathcal{M} \in \mathfrak{E}(X)$, we have $\mathcal{M} \bullet \mathcal{M}_\emptyset = \mathcal{M}$ which implies by definition that $\mathcal{M}_\emptyset \leq \mathcal{M}$. Similarily, it is easy to see that for all $\mathcal{M} \in \mathfrak{E}(X)$, we have $\mathcal{M} \bullet (X, \emptyset) = (X, \emptyset)$ which implies by definition that $\mathcal{M} \leq (X, \emptyset)$.

Part 2: $\mathfrak{E}_{\text{fin}}(X)$, is a sublattice of $\mathfrak{E}(X)$ with the same bottom and no top. It is clear that for every $\mathcal{M}_1 \in \mathfrak{E}_{\text{fin}}(X)$ and $\mathcal{M}_2 \in \mathfrak{E}(X)$ such that $\mathcal{M}_2 \leq \mathcal{M}_1$, we have that $\mathcal{M}_2 \in \mathfrak{E}_{\text{fin}}(X)$. It follows that $\mathfrak{E}_{\text{fin}}(X)$ is a sublattice of $\mathfrak{E}(X)$ with $\mathcal{M}_\emptyset$ as bottom. Moreover, if follows from the definition of $\bullet$ that if $\mathfrak{E}_{\text{fin}}(X)$ with X infinite had a top $(\mathcal{F}, \mathcal{P})$, we would have $S \subseteq \mathcal{F}$ for all $S \subseteq_{\text{fin}} X$. This means that $\mathcal{F}$ should be equal to X, which is not possible ~~since X is infinite.~~ .

Part 3: $\mathfrak{E}_{\text{eql}}(X)$ is a bounded sublattice of $\mathfrak{E}(X)$ with the same top and $(X, 2^X)$ as bottom. It is clear that for every $\mathcal{M}_1, \mathcal{M}_2 \in \mathfrak{E}_{\text{eql}}(X)$ we have that $\mathcal{M}_1 \bullet \mathcal{M}_2 \in \mathfrak{E}_{\text{eql}}(X)$ and $\mathcal{M}_1 \star \mathcal{M}_2 \in \mathfrak{E}_{\text{eql}}(X)$. It follows that $\mathfrak{E}_{\text{fin}}(X)$ is a sublattice of $\mathfrak{E}(X)$ with $(X, \emptyset)$ as top. Finally, it is easy to see that $(X, 2^X) \in \mathfrak{E}_{\text{eql}}(X)$ and that for all $\mathcal{M}_1 \in \mathfrak{E}_{\text{eql}}(X)$, we have $(X, 2^X) \bullet \mathcal{M} = \mathcal{M}$. □

@MICHAEL: please update according to the statement.

3.3 On Components and Interfaces

Theorem 2 (Interfaces are components). *If $\mathcal{M}_1 \preceq \mathcal{M}_2$ then $\mathcal{M}_1 \leq \mathcal{M}_2$.*

Proof. Immediate by Definition 6 and Lemma 1. □

Lemma 2 (Monotonocity properties of the feature model slice operator). *For all $\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2 \subseteq X$ and $\mathcal{M}, \mathcal{M}_1, \mathcal{M}_2 \in \mathfrak{E}(X)$*

1. *If $\mathcal{F}_1 \subseteq \mathcal{F}_2$ then $\Pi_{\mathcal{F}_1}(\mathcal{M}) \preceq \Pi_{\mathcal{F}_2}(\mathcal{M})$.*
2. *If $\mathcal{F}_1 \subseteq \mathcal{F}_2$ then $\Pi_{\mathcal{F}_1}(\mathcal{M}) \leq \Pi_{\mathcal{F}_2}(\mathcal{M})$.*
3. *If $\mathcal{M}_1 \preceq \mathcal{M}_2$ then $\Pi_{\mathcal{F}}(\mathcal{M}_1) \preceq \Pi_{\mathcal{F}}(\mathcal{M}_2)$.*
4. *If $\mathcal{M}_1 \leq \mathcal{M}_2$ then $\Pi_{\mathcal{F}}(\mathcal{M}_1) \leq \Pi_{\mathcal{F}}(\mathcal{M}_2)$.*

Proof. 1. Clearly $\Pi_{\mathcal{F}_1}(\mathcal{M}) \bullet \Pi_{\mathcal{F}_2}(\mathcal{M}) = \Pi_{\mathcal{F}_2}(\mathcal{M})$. Thus the proof follows by Definition 6.

2. Immediate by Lemma 2.1 and Theorem 2.
3. By Definition 6, we have that $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\mathcal{P}_1 = \mathcal{P}_2 |_{\mathcal{F}_1}$. Consequently, for all $\mathcal{F} \subseteq X$, we have $(\mathcal{F}_1 \cap \mathcal{F}) \subseteq (\mathcal{F}_2 \cap \mathcal{F})$ and $\mathcal{P}_1 |_{\mathcal{F}} = \mathcal{P}_2 |_{\mathcal{F}_1} |_{\mathcal{F}}$. Still, $\Pi_{\mathcal{F}}(\mathcal{M}_1) \preceq \Pi_{\mathcal{F}}(\mathcal{M}_2)$ by Definition 6.
4. By Lemma 1, we have that $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\mathcal{P}_1 \supseteq \mathcal{P}_2 |_{\mathcal{F}_1}$. Consequently, for all $\mathcal{F} \subseteq X$, we have $(\mathcal{F}_1 \cap \mathcal{F}) \subseteq (\mathcal{F}_2 \cap \mathcal{F})$ and $\mathcal{P}_1 |_{\mathcal{F}} \supseteq \mathcal{P}_2 |_{\mathcal{F}_1} |_{\mathcal{F}}$. Still, $\Pi_{\mathcal{F}}(\mathcal{M}_1) \leq \Pi_{\mathcal{F}}(\mathcal{M}_2)$ by Lemma 1.

We remark that Lemma 2.3 and Theorem 2 do not imply Lemma 2.4.

Theorem 3 (Algebraic properties of the feature model slice operator). *For all $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3 \in \mathfrak{E}(X)$ and $\mathcal{F}_4, \mathcal{F}_5 \subseteq X$, we have*

$\leq$-Monotonicity. *If $\mathcal{M}_1 \leq \mathcal{M}_2$ and $\mathcal{F}_4 \subseteq \mathcal{F}_5$, then $\Pi_{\mathcal{F}_4}(\mathcal{M}_1) \leq \Pi_{\mathcal{F}_5}(\mathcal{M}_2)$.*
$\preceq$-Monotonicity. *If $\mathcal{M}_1 \preceq \mathcal{M}_2$ and $\mathcal{F}_4 \subseteq \mathcal{F}_5$, then $\Pi_{\mathcal{F}_4}(\mathcal{M}_1) \preceq \Pi_{\mathcal{F}_5}(\mathcal{M}_2)$.*
Commutativity. *$\Pi_{\mathcal{F}_4}(\Pi_{\mathcal{F}_5}(\mathcal{M}_3)) = \Pi_{\mathcal{F}_5}(\Pi_{\mathcal{F}_4}(\mathcal{M}_3))$.*

Proof. **$\leq$-Monotonicity.** First, it is easy to see that for all $\mathcal{M} \in \mathfrak{E}(X)$, we have $\Pi_{\mathcal{F}_4}(\mathcal{M}) \bullet \Pi_{\mathcal{F}_5}(\mathcal{M}) = \Pi_{\mathcal{F}_5}(\mathcal{M})$. We can thus apply Lemma 1 to obtain that $\Pi_{\mathcal{F}_4}(\mathcal{M}) \leq \Pi_{\mathcal{F}_5}(\mathcal{M})$. Moreover, by Lemma 1, we have that $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\mathcal{P}_1 \supseteq \mathcal{P}_2 |_{\mathcal{F}_1}$. It is thus clear that for all $\mathcal{F} \subseteq X$, we have $(\mathcal{F}_1 \cap \mathcal{F}) \subseteq (\mathcal{F}_2 \cap \mathcal{F})$ and $\mathcal{P}_1 |_{\mathcal{F}} \supseteq \mathcal{P}_2 |_{\mathcal{F}_1} |_{\mathcal{F}}$, which implies, by Lemma 1, that $\Pi_{\mathcal{F}}(\mathcal{M}_1) \leq \Pi_{\mathcal{F}}(\mathcal{M}_2)$. We can then conclude by transitivity of $\leq$:

$$\Pi_{\mathcal{F}_4}(\mathcal{M}_1) \leq \Pi_{\mathcal{F}_5}(\mathcal{M}_1) \leq \Pi_{\mathcal{F}_5}(\mathcal{M}_2)$$

By Lemma 2.2 and Lemma 2.4.

$\preceq$-Monotonicity. Since $\mathcal{M}_1 \preceq \mathcal{M}_2$, there exists by definition $\mathcal{F} \subseteq X$ such that $\mathcal{M}_1 = \Pi_{\mathcal{F}}(\mathcal{M}_2)$. It follows that

$$\begin{aligned}
\Pi_{\mathcal{F}_4}(\mathcal{M}_1) &= \Pi_{\mathcal{F}_4}(\Pi_{\mathcal{F}}(\mathcal{M}_2)) \\
&= \Pi_{\mathcal{F}_4 \cap \mathcal{F}}(\mathcal{M}_2) \\
&= \Pi_{\mathcal{F}_4 \cap \mathcal{F}_5 \cap \mathcal{F}}(\mathcal{M}_2) \qquad \text{since } \mathcal{F}_4 \subseteq \mathcal{F}_5 \\
&= \Pi_{\mathcal{F}_4 \cap \mathcal{F}}(\Pi_{\mathcal{F}_5}(\mathcal{M}_2))
\end{aligned}$$

It follows, by definition, that $\Pi_{\mathcal{F}_4}(\mathcal{M}_1) \preceq \Pi_{\mathcal{F}_5}(\mathcal{M}_2)$. By Lemma 2.1 and Lemma 2.3.

Commutativity. In accord to to Definition 6, it is sufficient to observe that $\Pi_{\mathcal{F}_4}(\Pi_{\mathcal{F}_5}(\mathcal{M}_3)) = \Pi_{\mathcal{F}_4 \cup \mathcal{F}_5}(\mathcal{M}_3) = \Pi_{\mathcal{F}_5}(\Pi_{\mathcal{F}_4}(\mathcal{M}))$ holds. □

4 Propositional Characterization of Feature Models

In Section 4.1 we provide a formal account of the relation between the propositional representation and the extensional representation of feature models (cf. Sect. 2.1). Then, in Section 4.2, we provide a propositional characterization for the component relation ($\leq$), for the composition ($\bullet$) and the meet ($\star$) operations, for the feature models $\mathcal{M}_X$ (the bottom of the lattice $\mathfrak{E}(X)$) and $\mathcal{M}^X$ (the top of the lattice $\mathfrak{E}(X)$), and for the complement operation ($\bar{\ }$). Finally, in Section 4.3, we provide a propositional characterization for the slice operator (Π_Y) and for the interface relation ($\preceq$).

4.1 Relating Extensional and Propositional Feature Models

As stated at the beginning of Sect. 3.2, in our theoretical development we consider also feature models with infinitely many features and products, where each product may have infinitely many features. The following definition introduces a notion for three different sets of propositional feature models (see Definition 1) over a set of features (cf. Definition 8).

Definition 9 (Sets of propositional feature models over a set of features). *Let X be a set of features. We denote:*

- $\mathfrak{P}(X)$ *the set of the propositional feature models* $(\mathcal{F}, \phi)$ *such that* $\mathcal{F} \subseteq X$*;*
- $\mathfrak{P}_{\mathrm{fin}}(X)$ *the subset of the finite elements of* $\mathfrak{P}(X)$*, i.e.,* $(\mathcal{F}, \phi)$ *such that* $\mathcal{F} \subseteq_{\mathrm{fin}} X$*; and*
- $\mathfrak{P}_{\mathrm{eql}}(X)$ *the subset of elements of* $\mathfrak{P}(X)$ *that have exactly the features* X*, i.e.,* $(\mathcal{F}, \phi)$ *such that* $\mathcal{F} = X$*.*

We denote $\mathrm{ftrs}(\phi)$ the set of features occurring in a propositional formula ϕ, and (as usual) we say that ϕ is *ground* whenever $\mathrm{ftrs}(\phi)$ is empty. We recall that an *interpretation* (a.k.a. truth assignment or valuation) $\mathcal{I}$ is a function which maps propositional logic variables to true or false [4, 7]. As usual, we denote $\mathrm{dom}(\mathcal{I})$ the domain of an interpretation $\mathcal{I}$ and we write $\mathcal{I} \models \phi$ to mean that the propositional formula ϕ is true under the interpretation $\mathcal{I}$ (i.e., $\mathrm{ftrs}(\phi) \subseteq \mathrm{dom}(\mathcal{I})$ and the ground formula obtained from ϕ by replacing each feature x occurring in ϕ by $\mathcal{I}(x)$ evaluates to true). We write $\models \phi$ to mean that ϕ is valid (i.e., it evaluates to true under all the interpretations $\mathcal{I}$ such that $\mathrm{ftrs}(\phi) \subseteq \mathrm{dom}(\mathcal{I})$), we write $\phi_1 \models \phi_2$ to mean that ϕ_2 is is a logical consequence of ϕ_1 (i.e., for all interpretations $\mathcal{I}$ with $\mathrm{ftrs}(\phi_1) \cup \mathrm{ftrs}(\phi_2) \subseteq \mathrm{dom}(\mathcal{I})$, if $\mathcal{I} \models \phi_1$ then $\mathcal{I} \models \phi_2$), and we write $\phi_1 \equiv \phi_2$ to mean that ϕ_1 and ϕ_2 are logically equivalent (i.e., they are satisfied by exactly the same interpretations with domain including $\mathrm{ftrs}(\phi_1) \cup \mathrm{ftrs}(\phi_2)$). We recall that:

- $\mathcal{I}_1$ is *included* in $\mathcal{I}_2$, denoted $\mathcal{I}_1 \subseteq \mathcal{I}_2$, whenever $\mathrm{dom}(\mathcal{I}_1) \subseteq \mathrm{dom}(\mathcal{I}_2)$ and $\mathcal{I}_1(x) = \mathcal{I}_2(x)$, for all $x \in \mathrm{dom}(\mathcal{I}_1)$;
- $\mathcal{I}_1$ and $\mathcal{I}_2$ are *compatible* whenever $\mathcal{I}_1(x) = \mathcal{I}_2(x)$, for all $x \in \mathrm{dom}(\mathcal{I}_1) \cap \mathrm{dom}(\mathcal{I}_2)$; and
- if $\mathcal{I}_1 \models \phi$ then its restriction $\mathcal{I}_0$ to $\mathrm{ftrs}(\phi)$ is such that $\mathcal{I}_0 \models \phi$ and, for all interpretations $\mathcal{I}_2$ such that $\mathcal{I}_0 \subseteq \mathcal{I}_2$, it holds that $\mathcal{I}_2 \models \phi$.

The following definition gives a name to the interpretations that represent the products of the feature models with a given set of features.

Definition 10 (Interpretation representing a product). *Let $(\mathcal{F}, \mathcal{P})$ be an extensional feature model and $p \in \mathcal{P}$. The interpretation that* represents *the product p, denoted by $\mathcal{I}_p^{\mathcal{F}}$, is the interpretation with domain $\mathcal{F}$ such that:*

$$\mathcal{I}_p^{\mathcal{F}}(x) = \begin{cases} \mathsf{true} & \text{if } x \in p, \\ \mathsf{false} & \text{if } x \in \mathcal{F} \setminus p. \end{cases}$$

The following definition gives a name to the mapping that associates each propositional feature model to its extensional representation.

Definition 11 (The ext mapping). *Let $(\mathcal{F}, \phi) \in \mathfrak{P}(X)$. We denote $\texttt{ext}((\mathcal{F}, \phi))$ (or $\texttt{ext}(\mathcal{F}, \phi)$, for short) the extensional feature model $(\mathcal{F}, \mathcal{P}) \in \mathfrak{E}(X)$ such that $\mathcal{P} = \{p \mid p \subseteq \mathcal{F}$ and $\mathcal{I}_p^{\mathcal{F}} \models \phi\}$. In particular, $\texttt{ext}$ maps $\mathfrak{P}_{\text{fin}}(X)$ to $\mathfrak{E}_{\text{fin}}(X)$, and maps $\mathfrak{P}_{\text{eql}}(X)$ to $\mathfrak{E}_{\text{eql}}(X)$.*

We denote $\equiv$ the equivalence relation over feature models defined by: $(\mathcal{F}_1, \phi_1) \equiv (\mathcal{F}_2, \phi_2)$ if and only if both $\mathcal{F}_1 = \mathcal{F}_2$ and $\phi_1 \equiv \phi_2$. We write $[\mathfrak{P}(X)]$, $[\mathfrak{P}_{\text{fin}}(X)]$ and $[\mathfrak{P}_{\text{eql}}(X)]$ as short for the quotient sets $\mathfrak{P}(X)/\equiv$, $\mathfrak{P}_{\text{fin}}(X)/\equiv$ and $\mathfrak{P}_{\text{eql}}(X)/\equiv$, respectively.

Note that, if X has infinitely many elements and $(\mathcal{F}, \phi) \in \mathfrak{P}(X)$, then $\mathcal{F}$ may contain infinite many features, while the propositional formula ϕ is syntactically finite (cf. Definition 1). Moreover, $\mathfrak{P}_{\text{fin}}(X)$ has infinitely many elements (even when X if finite). It is also worth observing that, if X is finite, then $\mathfrak{P}(X)$ and $\mathfrak{P}_{\text{fin}}(X)$ are the same and the quotient set $[\mathfrak{P}_{\text{fin}}(X)]$ is finite.

Note that the mapping $\texttt{ext}$, for all $\Phi_1, \Phi_2 \in \mathfrak{P}(X)$, we have that: $\texttt{ext}(\Phi_1) = \texttt{ext}(\Phi_2)$ if and only if $\Phi_1 \equiv \Phi_2$.

All the finite feature models have a propositional representation, i.e., if $(\mathcal{F}, \mathcal{P}) \in \mathfrak{E}_{\text{fin}}(X)$, then there exists $(\mathcal{F}, \phi) \in \mathfrak{P}_{\text{fin}}(X)$ such that $\texttt{ext}(\mathcal{F}, \phi) = (\mathcal{F}, \mathcal{P})$. Take, for instance, the formula in disjunctive normal form $\phi = \bigvee_{p \in \mathcal{P}} \big((\wedge_{f \in p} f) \wedge (\wedge_{f \in \mathcal{F} \setminus p} \neg f)\big)$. Given $[\Phi_1]$, $[\Phi_2] \in [\mathfrak{P}(X)]$, we define (with an abuse of notation) $\texttt{ext}([\Phi_1]) = \texttt{ext}(\Phi_1)$. Then, we have that $\texttt{ext}$ is an injection from $[\mathfrak{P}(X)]$ to $\mathfrak{E}(X)$, an injection from $[\mathfrak{P}_{\text{eql}}(X)]$ to $\mathfrak{E}_{\text{eql}}(X)$, and a bijection from $[\mathfrak{P}_{\text{fin}}(X)]$ to $\mathfrak{E}_{\text{fin}}(X)$.

Instead, as showed by the following example, if X has infinitely many elements, then there are feature models in $\mathfrak{E}(X) \setminus \mathfrak{E}_{\text{fin}}(X)$ that have no propositional representation.

Example 6 (Extensional feature models without a propositional representation). Consider the natural numbers as features. Then the extensional feature models $(\mathbb{N}, \{\{3\}\})$, $(\mathbb{N}, \{\{n \mid n \text{ is even}\}\})$ and $(\mathbb{N}, \{\{n\} \mid n \text{ is even}\})$ have no propositional representation.

Note that, if $\phi \not\equiv \mathsf{false}$, then $(\mathcal{F}, \phi)$ may not constraint the value of the features in $\mathcal{F} \setminus \text{ftrs}(\phi)$. In particular, we have that $\texttt{ext}(\mathcal{F}, \phi) = \texttt{ext}(\text{ftrs}(\phi), \phi) \bullet (\mathcal{F} \setminus \text{ftrs}(\phi), 2^{\mathcal{F} \setminus \text{ftrs}(\phi)})$. Therefore (since the set $\text{ftrs}(\phi)$ is finite) we have that, if X has infinitely many elements, then there are infinitely many elements of $\mathfrak{P}(X) \setminus \mathfrak{P}_{\text{fin}}(X)$ that do not have a propositional representation.

4.2 Propositional Characterization of the Lattices of Feature Models

The following theorem states that the feature model component relation $\leq$ corresponds to (the converse of) logical consequence.

Theorem 4 (Propositional characterization of the relation $\leq$). *Given $\Phi_1 = (\mathcal{F}_1, \phi_1)$ and $\Phi_2 = (\mathcal{F}_2, \phi_2)$ in $\mathfrak{P}(X)$, we write (with an abuse of notation) $\Phi_1 \leq \Phi_2$ to mean that both $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\phi_2 \models \phi_1$ hold. Then: $\mathtt{ext}(\Phi_1) \leq \mathtt{ext}(\Phi_2)$ holds if and only $\Phi_1 \leq \Phi_2$ holds.*

Proof. We have: $(\mathcal{F}_1, \mathcal{P}_1) = \mathtt{ext}(\Phi_1) \leq \mathtt{ext}(\Phi_2) = (\mathcal{F}_2, \mathcal{P}_2)$
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\mathcal{P}_1 \supseteq \mathcal{P}_2|_{\mathcal{F}_1}$ (by Lemma 1)
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\{p_1 \mid \mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1\} \supseteq \{p_2 \cap \mathcal{F}_1 \mid \mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_2\}$
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and, for all $p \in \mathcal{P}_2$, $\mathcal{I}_p^{\mathcal{F}_2} \models \phi_2$ implies $\mathcal{I}_p^{\mathcal{F}_2} \models \phi_1$
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\phi_2 \models \phi_1$
iff $\Phi_1 \leq \Phi_2$. □

The following theorem provides a formal account of the fact that the feature model composition operator $\bullet$ corresponds to propositional conjunction (cf. Sect. 2.2).

Theorem 5 (Propositional characterization of the operator $\bullet$). *Given $\Phi_1 = (\mathcal{F}_1, \phi_1)$ and $\Phi_2 = (\mathcal{F}_2, \phi_2)$ in $\mathfrak{P}(X)$, we define (with an abuse of notation): $\Phi_1 \bullet \Phi_2 = (\mathcal{F}_1 \cup \mathcal{F}_2, \phi_1 \wedge \phi_2)$. Then: $\mathtt{ext}(\Phi_1) \bullet \mathtt{ext}(\Phi_2) = \mathtt{ext}(\Phi_1 \bullet \Phi_2)$.*

Proof. Let $\mathtt{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$, for $i = 1, 2$.
$\mathtt{ext}(\Phi_1) \bullet \mathtt{ext}(\Phi_2) = (\mathcal{F}_3, \mathcal{P}_3)$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cup \mathcal{F}_2$ and iff $\mathcal{P}_3 = \{p_1 \cup p_2 \mid \mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1, \mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_1, p_1 \cap \mathcal{F}_2 = p_2 \cap \mathcal{F}_1\}$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathcal{P}_3 = \{p \mid p_1 \cup p_2 \subseteq p$ and $\mathcal{I}_p^X \models \phi_1, \mathcal{I}_p^X \models \phi_2\}$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathcal{P}_3 = \{p \mid \mathcal{I}_p^{\mathcal{F}_3} \models \phi_1 \wedge \phi_2\}$
iff $(\mathcal{F}_3, \mathcal{P}_3) = \mathtt{ext}(\Phi_1 \bullet \Phi_2)$. □

In order to provide a propositional characterization of the meet operator $\star$ (introduced in Theorem 1), we introduce the following auxiliary notation (where Y is a finite set of features and ϕ a propositional formula over features):

$$(\bigvee_Y \phi) = \begin{cases} \phi & \text{if } Y = \emptyset, \\ (\bigvee_{Y - \{x\}} (\phi[x := \mathsf{true}]) \vee (\phi[x := \mathsf{false}])) & \text{otherwise.} \end{cases}$$

Theorem 6 (Propositional characterization of the operator $\star$). *Given $\Phi_1 = (\mathcal{F}_1, \phi_1)$ and $\Phi_2 = (\mathcal{F}_2, \phi_2)$ in $\mathfrak{P}(X)$, we define (with an abuse of notation):*

$$\Phi_1 \star \Phi_2 = \big(\mathcal{F}_1 \cap \mathcal{F}_2, (\bigvee_{\mathrm{ftrs}(\phi_1) \setminus \mathcal{F}_2} \phi_1) \vee (\bigvee_{\mathrm{ftrs}(\phi_2) \setminus \mathcal{F}_1} \phi_2)\big).$$

Then: $\mathtt{ext}(\Phi_1) \star \mathtt{ext}(\Phi_2) = \mathtt{ext}(\Phi_1 \star \Phi_2)$.

Proof. Let $\mathtt{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$ for $i = 1, 2$.
Since $\mathtt{ext}(\mathcal{F}_1, \phi_1) \star \mathtt{ext}(\mathcal{F}_2, \phi_2) = (\mathcal{F}_1 \cap \mathcal{F}_2, \mathcal{P}_1|_{\mathcal{F}_2} \cup \mathcal{P}_2|_{\mathcal{F}_1})$, we have that:

$\mathtt{ext}(\Phi_1) \star \mathtt{ext}(\Phi_2) = (\mathcal{F}_3, \mathcal{P}_3)$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cap \mathcal{F}_2$ and $\mathcal{P}_3 = \{p_1 \cap \mathcal{F}_2 \mid \mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1\} \cup \{p_2 \cap \mathcal{F}_1 \mid \mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_2\}$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cap \mathcal{F}_2$ and $\mathcal{P}_3 = \{p_1 \cap \mathcal{F}_3 \mid \mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1\} \cup \{p_2 \cap \mathcal{F}_3 \mid \mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_2\}$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cap \mathcal{F}_2$ and $\mathcal{P}_3 = \mathcal{P}_1|_{\mathcal{F}_3} \cup \mathcal{P}_2|_{\mathcal{F}_3}$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cap \mathcal{F}_2$ and, $p \in \mathcal{P}_3$ implies
either $\exists p_1$ s.t. $p = p_1 \cap \mathcal{F}_3$ and $\mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1$ or $\exists p_2$ s.t. $p = p_2 \cap \mathcal{F}_3$ and $\mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_2$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cap \mathcal{F}_2$ and, $p \in \mathcal{P}_3$ implies
either $\mathcal{I}_p^{\mathcal{F}_3} \models (\bigvee_{\mathrm{ftrs}(\phi_1)\setminus\mathcal{F}_2} \phi_1)$ or $\mathcal{I}_p^{\mathcal{F}_3} \models (\bigvee_{\mathrm{ftrs}(\phi_2)\setminus\mathcal{F}_1} \phi_2)$
iff $\mathcal{F}_3 = \mathcal{F}_1 \cap \mathcal{F}_2$ and, $p \in \mathcal{P}_3$ implies $\mathcal{I}_p^{\mathcal{F}_3} \models (\bigvee_{\mathrm{ftrs}(\phi_1)\setminus\mathcal{F}_2} \phi_1) \vee (\bigvee_{\mathrm{ftrs}(\phi_2)\setminus\mathcal{F}_1} \phi_2)$
iff $(\mathcal{F}_3, \mathcal{P}_3) = \mathtt{ext}(\Phi_1 \star \Phi_2)$. □

The following theorem states that the feature models of the form $\mathcal{M}_\mathcal{F} = (\mathcal{F}, 2^\mathcal{F})$ and $\mathcal{M}^\mathcal{F} = (\mathcal{F}, \emptyset)$ correspond to true and false, respectively—recall that (see Theorem 1) $\mathcal{M}_\emptyset$ is the bottom of the lattices $(\mathfrak{E}(X), \leq)$ and $(\mathfrak{E}_{\mathrm{fin}}(X), \leq)$, while $\mathcal{M}_X$ is the bottom of the Boolean lattice $(\mathfrak{E}_{\mathrm{eql}}(X), \leq)$, and $\mathcal{M}^X$ is the top of the lattice $(\mathfrak{E}(X), \leq)$ and of the Boolean lattice $(\mathfrak{E}_{\mathrm{eql}}(X), \leq)$ and, if X is finite, of the lattice $(\mathfrak{E}_{\mathrm{fin}}(X), \leq)$.

Theorem 7 (Propositional characterization of the feature models $\mathcal{M}_\mathcal{F}$ and $\mathcal{M}^\mathcal{F}$). *Let $(\mathcal{F}, \phi) \in \mathfrak{P}(X)$.*

1. *$\mathtt{ext}(\mathcal{F}, \phi) = \mathcal{M}_\mathcal{F} = (\mathcal{F}, 2^\mathcal{F})$ if and only if $\phi \equiv \mathsf{true}$.*
2. *$\mathtt{ext}(\mathcal{F}, \phi) = \mathcal{M}^\mathcal{F} = (\mathcal{F}, \emptyset)$ if and only if $\phi \equiv \mathsf{false}$.*

Proof. *1.* Immediate, because true is satisfied by all interpretations. *2.* Immediate, because no interpretation satisfies false. □

The following theorem shows that the feature model complement operator $\overline{}$ (introduced in Theorem 1) corresponds to logical negation.

Theorem 8 (Propositional characterization of the operator $\overline{}$). *Given $\Phi = (\mathcal{F}, \phi)$ in $\mathfrak{P}(X)$, we define (with an abuse of notation): $\overline{\Phi} = (\mathcal{F}, \neg\phi)$. Then $\overline{\mathtt{ext}(\Phi)} = \mathtt{ext}(\overline{\Phi})$.*

Proof. Straightforward. □

The following lemma introduces a novel feature model operator, that we denote $+$, which corresponds to logical disjunction.

Lemma 3 (Propositional characterization of the operator $+$). *Given two sets Y and Z, we define: $Y \Cup Z = \{y \cup z \mid y \in Y, z \in Z\}$. Given two feature models $\mathcal{M}_1 = (\mathcal{F}_1, \mathcal{P}_1)$ and $\mathcal{M}_2 = (\mathcal{F}_2, \mathcal{P}_2)$ in $\mathfrak{E}(X)$, we define: $\mathcal{M}_1 + \mathcal{M}_2 = (\mathcal{F}_1 \cup \mathcal{F}_2, (\mathcal{P}_1 \Cup 2^{(\mathcal{F}_2 \setminus \mathcal{F}_1)}) \cup (\mathcal{P}_2 \Cup 2^{(\mathcal{F}_1 \setminus \mathcal{F}_2)}))$. Given $\Phi_1 = (\mathcal{F}_1, \phi_1)$ and $\Phi_2 = (\mathcal{F}_2, \phi_2)$ in $\mathfrak{P}(X)$, we define (with an abuse of notation): $\Phi_1 + \Phi_2 = (\mathcal{F}_1 \cup \mathcal{F}_2, \phi_1 \vee \phi_2)$. Then: $\mathtt{ext}(\Phi_1) + \mathtt{ext}(\Phi_2) = \mathtt{ext}(\Phi_1 + \Phi_2)$.*

Proof. Let $\mathtt{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$ for $i = 1, 2$. We have that
$\mathtt{ext}(\Phi_1) + \mathtt{ext}(\Phi_2) = (\mathcal{F}_3, \mathcal{P}_3)$
iff $\mathcal{P}_3 = \mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathcal{P}_3 = \{p_1 \Cup 2^{(\mathcal{F}_2 \setminus \mathcal{F}_1)}) \mid \mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1\} \cup \{p_2 \Cup 2^{(\mathcal{F}_1 \setminus \mathcal{F}_2)} \mid \mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_2\}$
iff $\mathcal{P}_3 = \mathcal{F}_1 \cup \mathcal{F}_2$ and, $p \in \mathcal{P}_3$ implies
either $p \in \{p_1 \Cup 2^{(\mathcal{F}_2 \setminus \mathcal{F}_1)}) \mid \mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1\}$ or $p \in \{p_2 \Cup 2^{(\mathcal{F}_1 \setminus \mathcal{F}_2)} \mid \mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_2\}$
iff $\mathcal{P}_3 = \mathcal{F}_1 \cup \mathcal{F}_2$ and, $p \in \mathcal{P}_3$ implies either $\mathcal{I}_p^{\mathcal{F}_3} \models \phi_1$ or $\mathcal{I}_p^{\mathcal{F}_3} \models \phi_2$
iff $\mathcal{P}_3 = \mathcal{F}_1 \cup \mathcal{F}_2$ and, $p \in \mathcal{P}_3$ implies $\mathcal{I}_{p \cap \mathcal{F}_1}^{\mathcal{F}_3} \models \phi_1 \vee \phi_2$
iff $(\mathcal{F}_3, \mathcal{P}_3) = \mathtt{ext}(\Phi_1 + \Phi_2)$. □

The following lemma shed some light on the Boolean lattice $\mathfrak{E}_{\mathrm{eql}}(X)$.

Lemma 4 (The operators $\star$, $+$ and $\bullet$ on $\mathfrak{E}_{\mathrm{eql}}(X)$). *Given two feature models $\mathcal{M}_1 = (X, \mathcal{P}_1)$ and $\mathcal{M}_2 = (X, \mathcal{P}_2)$ in $\mathfrak{E}_{\mathrm{eql}}(X)$, we have that:*

1. $\mathcal{M}_1 \bullet \mathcal{M}_2 = (X, \mathcal{P}_1 \cap \mathcal{P}_2)$, *and*
2. $\mathcal{M}_1 \star \mathcal{M}_2 = \mathcal{M}_1 + \mathcal{M}_2 = (X, \mathcal{P}_1 \cup \mathcal{P}_2)$.

Proof. 1. According to the definition of $\bullet$ we have:
$\mathcal{M}_1 \bullet \mathcal{M}_2 = (X, \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 = p_2\}) = (X, \mathcal{P}_1 \cup \mathcal{P}_2)$.
2. Straightforward from the definitions of $\star$ and $+$. □

Given $[\Phi_1]$, $[\Phi_2] \in [\mathfrak{P}(X)]$, we define (with an abuse of notation): $[\Phi_1] \leq [\Phi_2]$ as $\Phi_1 \leq \Phi_2$, $[\Phi_1] \bullet [\Phi_2] = [\Phi_1 \bullet \Phi_2]$, $[\Phi_1] \star [\Phi_2] = [\Phi_1 \star \Phi_2]$, $[\Phi_1] + [\Phi_2] = [\Phi_1 + \Phi_2]$, and $\overline{[\Phi_1]} = [\overline{\Phi_1}]$. Recall that a *homomorphism* is a structure-preserving map between two algebraic structures of the same type (e.g., between two lattices), a *monomorphism* is a injective homomorphism, and an *isomorphism* is a bijective homomorphism.

Theorem 9 (ext is a lattice monomorphism).

- $([\mathfrak{P}(X)], \leq)$ *is a bounded lattice with join $\bullet$, meet $\star$, bottom $[(\emptyset, \mathsf{true})]$ and top $[(X, \mathsf{false})]$. Moreover,* **ext** *is a bounded lattice monomorphism from* $([\mathfrak{P}(X)], \leq)$ *to* $(\mathfrak{E}(X), \leq)$.
- *If X has infinitely many elements, then $[\mathfrak{P}_{\mathrm{fin}}(X)]$ is a sublattice of $[\mathfrak{P}(X)]$ with the same bottom and no top. Moreover,* **ext** *is a lattice isomorphism from $[\mathfrak{P}_{\mathrm{fin}}(X)]$ to $\mathfrak{E}_{\mathrm{fin}}(X)$.*
- $[\mathfrak{P}_{\mathrm{eql}}(X)]$ *is a sublattice of $[\mathfrak{P}(X)]$ which is a Boolean lattice with bottom $[(X, \mathsf{true})]$, same top of $[\mathfrak{P}(X)]$, complement $^-$, and where the meet behaves like $+$. Moreover,* **ext** *is a Boolean lattice monomorphism from $[\mathfrak{P}_{\mathrm{eql}}(X)]$ to $\mathfrak{E}_{\mathrm{eql}}(X)$, which is an isomorphism whenever X is finite.*

Proof. Straightforward from Theorems 1, 4-8 and Lemmas 3 and 4.

4.3 Propositional Characterization of Slices and Interfaces

The following theorem provides a propositional characterization of the slice operator.

Theorem 10 (Propositional characterization of the operator Π_Y). *Let $\Phi = (\mathcal{F}, \phi)$ be in $\mathfrak{P}(X)$. We define (with an abuse of notation): $\Pi_Y(\Phi) = (Y \cap \mathcal{F}, (\bigvee_{\mathrm{ftrs}(\phi)\setminus Y} \phi))$. Then: $\Pi_Y(\mathtt{ext}(\Phi)) = \mathtt{ext}(\Pi_Y(\Phi))$.*

Proof. We have: $\Pi_Y(\mathtt{ext}(\Phi)) = (\mathcal{F}_0, \mathcal{P}_0)$
iff $\mathcal{F}_0 = \mathcal{F} \cap Y$ and $\mathcal{P}_0 = \{p \mid \mathcal{I}_p^{\mathcal{F}} \models \phi\}|_Y$
iff $\mathcal{F}_0 = \mathcal{F} \cap Y$ and $\mathcal{P}_0 = \{p \cap Y \mid \mathcal{I}_p^{\mathcal{F}} \models \phi\}$
iff $\mathcal{F}_0 = \mathcal{F} \cap Y$ and $\mathcal{P}_0 = \{p \cap Y \mid \mathcal{I}_p^{\mathcal{F}\cap Y} \models \phi\}$
iff $\mathcal{F}_0 = \mathcal{F} \cap Y$ and, $p_0 \in \mathcal{P}_0$ implies $\mathcal{I}_{p_0}^{\mathcal{F}\cap Y} \models (\bigvee_{\mathrm{ftrs}(\phi)\setminus Y} \phi)$
iff $(\mathcal{F}_0, \mathcal{P}_0) = \mathtt{ext}(\Pi_Y(\Phi))$. $\square$

The following corollary provides a propositional characterization of the interface relation $\mathcal{M}_1 \preceq \mathcal{M}_2$ which, unsurprisingly, is the same of the interpretation of the slice operator $\mathcal{M}_1 = \Pi_Y(\mathcal{M}_2)$ when Y are the features of $\mathcal{M}_1$ (cf. Theorem 10 and Remark 1).

Corollary 1 (Propositional characterization of the relation $\preceq$). *Given $\Phi_1 = (\mathcal{F}_1, \phi_1)$ and $\Phi_2 = (\mathcal{F}_2, \phi_2)$ in $\mathfrak{P}(X)$, we write (with an abuse of notation) $\Phi_1 \preceq \Phi_2$ to mean that both both $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\phi_1 \equiv (\bigvee_{\mathrm{ftrs}(\phi_2)\setminus \mathcal{F}_1} \phi_2)$ hold. Then: $\mathtt{ext}(\Phi_1) \preceq \mathtt{ext}(\Phi_2)$ holds if and only $\Phi_1 \preceq \Phi_2$ holds.*

Proof. We have:
$(\mathcal{F}_1, \mathcal{P}_1) = \mathtt{ext}(\Phi_1) \preceq \mathtt{ext}(\Phi_2) = (\mathcal{F}_2, \mathcal{P}_2)$
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\mathcal{P}_1 = \mathcal{P}_2|_{\mathcal{F}_1}$ (by Definition 6)
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\{p_1 \mid \mathcal{I}_{p_1}^{\mathcal{F}_1} \models \phi_1\} = \{p_2 \cap \mathcal{F}_1 \mid \mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_2\}$
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and, for all $p \in \mathcal{P}_2$, both $\phi_2 \models \phi_1$ and $\phi_1 \models (\bigvee_{\mathrm{ftrs}(\phi_2)\setminus \mathcal{F}_1} \phi_2)$
iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\phi_1 \equiv (\bigvee_{\mathrm{ftrs}(\phi_2)\setminus \mathcal{F}_1} \phi_2)$
iff $\Phi_1 \preceq \Phi_2$. $\square$

5 Related Work

WORK IN PROGRESS.

Although the propositional representation of feature models is well known in the literature (see, e.g., Sect. 2.3 of Apel *et al.* [4]), we are not aware of any work that (as done in the present paper) formalizes this representation in the general case of feature models with infinitely many features and relates it with an algebraic characterization of feature model operators and relations. The starting point of the instigation presented in this paper is the feature model fragment relation, which is induced by the feature model composition operator considered in [25, 20] to investigate industrial-size configuration spaces. In the following we discuss related work on feature model composition operators and on feature model relations.

Feature-model composition operators are often investigated in connection with multi software product lines, which are sets of interdependent product lines [?,?,?,19]. Eichelberger and Schmid [?] present an overview of textual-modeling languages which support variability-model composition (like FAMILIAR [?], VELVET [24], TVL [13], VSL [?]) and discuss their support for composition, modularity, and evolution. Acher *et al.* [3] consider different feature-model composition operators together with possible implementations and discuss advantages and drawbacks.

The feature-model fragment relation introduced in this paper generalizes the feature-model interface relation introduced by Schröter *et al.* [25], which (see Remark 1) is closely related to the feature model slice opeator introduced by Acher *et al.* [2]. The work of Acher *et al.* [2] focuses on feature model decomposition. In subsequent work [?], Acher *et al.*use the slice operator in combination with a merge operator to address evolutionary changes for extracted variability models, focusing on detecting differences between feature-model versions during evolution.

===

Instead, Schröter *et al.* [25] study how feature model interfaces can be used to support evolution for a feature model composed from feature models fragments. Changes to fragments which do not affect their interfaces do not require the overall feature model to be rebuilt (by composing the fragments) in order to reanalyze it. Challenges encountered to support evolution in software product line engineering have previously been studied by Dhungana *et al.* [?]. They use interfaces to hide information in feature model fragments and save a merge history of fragments to give feedback and facilitate fragment maintenance. No automated analysis is considered. In contrast to this work on feature model interfaces for evolution, the

in our work is for efficient automated product discovery in huge feature models represented as interdependent feature model fragments.

Feature-model views [?,?,?] focus on a subset of the relevant features of a given feature model, similarly to feature-model interfaces. Different views regarding one master feature model are used to capture the needs of different stakeholders, so that a product of the master feature model can be identified based on the views' partial configurations. This work on multiple views to a product in a feature model is orthogonal to our work on an algebraic characterization of feature-model operations and relations.

, which targets the efficient configuration of systems comprising many interdependent configurable packages.

6 Conclusion

References

1. Acher, M., Collet, P., Lahire, P., France, R.B.: Comparing approaches to implement feature model composition. In: Kühne, T., Selic, B., Gervais, M., Terrier, F. (eds.) Proc. 6th European Conference on Modelling Foundations and Applications (ECMFA 2010). pp. 3–19 (2010)

2. Acher, M., Collet, P., Lahire, P., France, R.B.: Slicing feature models. In: 26th IEEE/ACM International Conference on Automated Software Engineering, (ASE), 2011. pp. 424–427 (2011). https://doi.org/10.1109/ASE.2011.6100089
3. Acher, M., Combemale, B., Collet, P., Barais, O., Lahire, P., France, R.B.: Composing your compositions of variability models. In: Moreira, A., Schätz, B., Gray, J., Vallecillo, A., Clarke, P.J. (eds.) Proc. 16th International Conference on Model-Driven Engineering Languages and Systems (MODELS 2013). pp. 352–369 (2013). https://doi.org/10.1007/978-3-642-41533-3_22, https://doi.org/10.1007/978-3-642-41533-3_22
4. Apel, S., Batory, D., Kstner, C., Saake, G.: Feature-Oriented Software Product Lines: Concepts and Implementation. Springer Publishing Company, Incorporated, 1st edn. (2016)
5. Apel, S., Batory, D.S., Kästner, C., Saake, G.: Feature-Oriented Software Product Lines: Concepts and Implementation. Springer (2013)
6. Batory, D.: Feature models, grammars, and propositional formulas. In: Proceedings of International Software Product Line Conference (SPLC). LNCS, vol. 3714, pp. 7–20. Springer (2005). https://doi.org/10.1007/11554844_3
7. Ben-Ari, M.: Mathematical Logic for Computer Science. Springer Publishing Company, Incorporated, 3rd edn. (2012)
8. Benavides, D., Segura, S., Ruiz-Cortés, A.: Automated analysis of feature models 20 years later: A literature review. Information Systems **35**(6), 615–636 (2010). https://doi.org/10.1016/j.is.2010.01.001, https://doi.org/10.1016/j.is.2010.01.001
9. Berger, T., Rublack, R., Nair, D., Atlee, J.M., Becker, M., Czarnecki, K., Wąsowski, A.: A survey of variability modeling in industrial practice. pp. 7:1–7:8 (2013)
10. Berger, T., She, S., Lotufo, R., Wąsowski, A., Czarnecki, K.: Variability modeling in the real: a perspective from the operating systems domain. pp. 73–82 (2010)
11. Blyth, T.: Lattices and Ordered Algebraic Structures. Universitext, Springer London (2005), https://books.google.it/books?id=jQYwCI__1mgC
12. Bošković, M., Mussbacher, G., Bagheri, E., Amyot, D., Gašević, D., Hatala, M.: Aspect-oriented feature models. In: Dingel, J., Solberg, A. (eds.) Proc. Models in Software Engineering - Workshops and Symposia at MODELS 2010. pp. 110–124 (2010)
13. Classen, A., Boucher, Q., Heymans, P.: A text-based approach to feature modelling: Syntax and semantics of tvl. Sci. Comput. Program. **76**(12), 1130 – 1143 (2011). https://doi.org/https://doi.org/10.1016/j.scico.2010.10.005
14. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (Jun 1970). https://doi.org/10.1145/362384.362685
15. Damiani, F., Lienhardt, M., Paolini, L.: A formal model for multi software product lines. Science of Computer Programming **172**, 203 – 231 (2019). https://doi.org/10.1016/j.scico.2018.11.005
16. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, 2 edn. (2002). https://doi.org/10.1017/CBO9780511809088
17. Foundation, G.: Gentoo linux (2019), https://gentoo.org, last visited, 2019-08-20
18. Holl, G., Grünbacher, P., Rabiser, R.: A systematic review and an expert survey on capabilities supporting multi product lines. Information & Software Technology **54**(8), 828–852 (2012). https://doi.org/10.1016/j.infsof.2012.02.002
19. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Tech. Rep. CMU/SEI-90-TR-21, Carnegie Mellon Software Engineering Institute (1990)

20. Lienhardt, M., Damiani, F., Donetti, S., Paolini, L.: Multi software product lines in the wild. In: Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems, VAMOS 2018, Madrid, Spain, February 7-9, 2018. pp. 89–96. ACM, New York, USA (2018). https://doi.org/10.1145/3168365.3170425, https://doi.org/10.1145/3168365.3170425
21. Lienhardt, M., Damiani, F., Johnsen, E.B., Mauro, J.: Lazy product discovery in huge configuration spaces. In: Proceedings of the 42th International Conference on Software Engineering. ICSE '20, ACM (2020). https://doi.org/10.1145/3377811.3380372
22. Lotufo, R., She, S., Berger, T., Czarnecki, K., Wąsowski, A.: Evolution of the linux kernel variability model. In: Proceedings of the 14th International Conference on Software Product Lines: Going Beyond. pp. 136–150. SPLC'10, Springer-Verlag, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15579-6_10
23. Mendonca, M., Wasowski, A., Czarnecki, K.: SAT-based analysis of feature models is easy. In: Muthig, D., McGregor, J.D. (eds.) Proceedings of the 13th International Software Product Line Conference. ACM International Conference Proceeding Series, vol. 446, pp. 231–240. ACM (2009)
24. Rosenmüller, M., Siegmund, N., Kästner, C., Rahman, S.S.U.: Modeling dependent software product lines. In: Proc. Workshop on Modularization, Composition and Generative Techniques for Product Line Engineering (McGPLE). pp. 13–18. Department of Informatics and Mathematics, University of Passau (2008)
25. Rosenmüller, M., Siegmund, N., Thüm, T., Saake, G.: Multi-dimensional variability modeling. pp. 11–20 (2011). https://doi.org/10.1145/1944892.1944894, http://doi.acm.org/10.1145/1944892.1944894
26. Schröter, R., Krieter, S., Thüm, T., Benduhn, F., Saake, G.: Feature-model interfaces: The highway to compositional analyses of highly-configurable systems. In: Proceedings of the 38th International Conference on Software Engineering. pp. 667–678. ICSE '16, ACM (2016). https://doi.org/10.1145/2884781.2884823
27. Schröter, R., Thüm, T., Siegmund, N., Saake, G.: Automated analysis of dependent feature models. pp. 9:1–9:5 (2013). https://doi.org/10.1145/2430502.2430515
28. Tartler, R., Lohmann, D., Sincero, J., Schröder-Preikschat, W.: Feature consistency in compile-time-configurable system software: facing the linux 10, 000 feature problem. In: Kirsch, C.M., Heiser, G. (eds.) Proc. 6th European Conference on Computer systems (EuroSys 2011). pp. 47–60 (2011)
29. Thüm, T., Apel, S., Kästner, C., Schaefer, I., Saake, G.: A classification and survey of analysis strategies for software product lines. ACM Comput. Surv. **47**(1), 6:1–6:45 (2014). https://doi.org/10.1145/2580950

A — — — Michael's OLD Section 3 — — —

Definition 12 (Feature Model). *A* Feature Model *$\mathcal{M}$ is a pair $(\mathcal{F},\mathcal{P})$ where $\mathcal{F}$ is a set and $\mathcal{P} \subseteq 2^{\mathcal{F}}$ is a set of products. $\mathcal{M}_\emptyset = (\emptyset, \{\emptyset\})$ is the* empty *feature model. For any set X, we write $\mathfrak{E}(X)$ to denote the set of all feature model $(\mathcal{F},\mathcal{P})$ such that $\mathcal{F} \subseteq X$.*

La prima delle operazioni qui sotto (viz. $\cup$) non è mai usata: sbaglio?

For any set X, we define the operators $\cup$ and $\bullet$ on $\mathfrak{E}(X)$ as follows:

$$(\mathcal{F}_x,\mathcal{P}_x) \cup (\mathcal{F}_y,\mathcal{P}_y) = (\mathcal{F}_x \cup \cancel{\mathcal{P}_x}\ \mathcal{F}_y, \mathcal{P}_x \cup \mathcal{P}_y)$$

$$(\mathcal{F}_x,\mathcal{P}_x) \bullet (\mathcal{F}_y,\mathcal{P}_y) = (\mathcal{F}_x \cup \mathcal{F}_y, \{p \cup q \mid p \in \mathcal{P}_x, q \in \mathcal{P}_y, p \cap \mathcal{F}_y = q \cap \mathcal{F}_x\})$$

Additionally, we define the binary relation $\leq$ on $\mathfrak{E}(X)$ such that:

$$\mathcal{M}_x \leq \mathcal{M}_y \Leftrightarrow \exists \mathcal{M}_z,\ \mathcal{M}_y = \mathcal{M}_x \bullet \mathcal{M}_z$$

i sup-inf sono comunemente chiamati (principal)-filters and ideals ed indicati come $\uparrow\mathcal{M}$ and $\downarrow\mathcal{M}$

Moreover, for any $\mathcal{M} \in \mathfrak{E}(X)$, we write $\sup(\mathcal{M}) = \{\mathcal{M}' \in \mathfrak{E}(X) \mid \mathcal{M} \leq \mathcal{M}'\}$ *and* $\inf(\mathcal{M}) = \{\mathcal{M}' \in \mathfrak{E}(X) \mid \mathcal{M}' \leq \mathcal{M}\}$.

Theorem 11. *For any set X, we have the following properties:*

- *$(\mathfrak{E}(X), \bullet, \mathcal{M}_\emptyset)$ is a positive, idempotent and commutative monoid. Moreover, this monoid is not cancellative.*
- *$(\mathfrak{E}(X), \leq)$ is a lattice with $\mathcal{M}_\emptyset$ as lower bound and such that for all $\mathcal{M}_x = (\mathcal{F}_x,\mathcal{P}_x), \mathcal{M}_y = (\mathcal{F}_y,\mathcal{P}_y) \in \mathfrak{E}(X)$:*
  - *$\mathcal{M}_x \sqcap \mathcal{M}_y = \mathcal{M}_x \bullet \mathcal{M}_y$ where $\sqcap$ should be sup/join/lub in accord to the proof!*
    **** La prova segue dal Theorem 2.2 [11, p.20] dove l'ordine è invertito, che mostra che : Every commutative idempotent semigroup can be ordered in such a way that it forms a meet semilattice. ****
  - *$\mathcal{M}_x \sqcup \mathcal{M}_y = (\mathcal{F}_x \cap \mathcal{F}_y, \{p_x \cap \mathcal{F}_y \mid p_x \in \mathcal{P}_x\} \cup \{p_y \cap \mathcal{F}_x \mid p_y \in \mathcal{P}_y\})$ where $\sqcup$ denotes a lub!?!!!! Indeed, we have an idempotent semiring ... and its induced lattice*

Proof. We prove this theorem in four distinct parts.
Part 1: $(\mathfrak{E}(X), \bullet, \mathcal{M}_\emptyset)$ is a monoid with the described properties. For any $(\mathcal{F}_x,\mathcal{P}_x), (\mathcal{F}_y,\mathcal{P}_y), (\mathcal{F}_z,\mathcal{P}_z) \in \mathfrak{E}(X)$, we have

- Associativity: gia dimostrato in [15, Appendix A, p.225]

$$\begin{aligned}
&(\mathcal{F}_x,\mathcal{P}_x) \bullet ((\mathcal{F}_y,\mathcal{P}_y) \bullet (\mathcal{F}_z,\mathcal{P}_z)) \\
&= (\mathcal{F}_x,\mathcal{P}_x) \bullet (\mathcal{F}_y \cup \mathcal{F}_z, \{p_y \cup p_z \mid p_y \in \mathcal{P}_y,\ p_z \in \mathcal{P}_z,\ p_y \cap \mathcal{F}_z = p_z \cap \mathcal{F}_y\}) \\
&= (\mathcal{F}_x \cup \mathcal{F}_y \cup \mathcal{F}_z, \{p_x \cup p_y \cup p_z \mid p_x \in \mathcal{P}_x,\ p_y \in \mathcal{P}_y,\ p_z \in \mathcal{P}_z, \\
&\qquad p_y \cap \mathcal{F}_z = p_z \cap \mathcal{F}_y,\ p_x \cap (\mathcal{F}_y \cup \mathcal{F}_z) = (p_y \cup p_z) \cap \mathcal{F}_x\}) \\
&= (\mathcal{F}_x \cup \mathcal{F}_y \cup \mathcal{F}_z, \{p_x \cup p_y \cup p_z \mid p_x \in \mathcal{P}_x,\ p_y \in \mathcal{P}_y,\ p_z \in \mathcal{P}_z, \\
&\qquad p_y \cap \mathcal{F}_z = p_z \cap \mathcal{F}_y,\ p_x \cap \mathcal{F}_y = p_y \cap \mathcal{F}_x,\ p_x \cap \mathcal{F}_z = p_z \cap \mathcal{F}_x\}) \\
&\qquad \text{as } \ p_x\cap(\mathcal{F}_y\cup\mathcal{F}_z)=(p_y\cup p_z)\cap\mathcal{F}_x \\
&\qquad \Leftrightarrow p_x\cap\mathcal{F}_z=(p_y\cap\mathcal{F}_z\cup p_z)\cap\mathcal{F}_x \wedge p_x\cap\mathcal{F}_y=(p_y\cup p_z\cap\mathcal{F}_y)\cap\mathcal{F}_x \\
&\qquad \Leftrightarrow p_x\cap\mathcal{F}_z=p_z\cap\mathcal{F}_x \wedge p_x\cap\mathcal{F}_y=p_y\cap\mathcal{F}_x \quad \text{due to } p_y \cap \mathcal{F}_z = p_z \cap \mathcal{F}_y \\
&= ((\mathcal{F}_x,\mathcal{P}_x) \bullet (\mathcal{F}_y,\mathcal{P}_y)) \bullet (\mathcal{F}_z,\mathcal{P}_z)
\end{aligned}$$

- Commutativity: gia dimostrato in [15, Appendix A, p.225]

$(\mathcal{F}_x, \mathcal{P}_x) \bullet (\mathcal{F}_y, \mathcal{P}_y)$
$= (\mathcal{F}_x \cup \mathcal{F}_y, \{p \cup q \mid p \in \mathcal{P}_x,\, q \in \mathcal{P}_y,\, p \cap \mathcal{F}_y = q \cap \mathcal{F}_x\})$
$= (\mathcal{F}_y \cup \mathcal{F}_x, \{q \cup p \mid p \in \mathcal{P}_x,\, q \in \mathcal{P}_y,\, p \cap \mathcal{F}_y = q \cap \mathcal{F}_x\})$
$= (\mathcal{F}_y, \mathcal{P}_y) \bullet (\mathcal{F}_x, \mathcal{P}_x)$

- Neutral Element: gia dimostrato in [15, Appendix A, p.225]

$(\mathcal{F}_x, \mathcal{P}_x) \bullet (\emptyset, \{\emptyset\})$
$= (\mathcal{F}_x \cup \emptyset, \{p \cup q \mid p \in \mathcal{P}_x,\, q \in \{\emptyset\},\, p \cap \emptyset = q \cap \mathcal{F}_x\})$
$= (\mathcal{F}_x, \{p \mid p \in \mathcal{P}_x,\, p \cap \emptyset = \emptyset\})$
$= (\mathcal{F}_x, \mathcal{P}_x)$

- Positivity: in presenza di elemento neutro è triviale: togliere?

$(\mathcal{F}_x, \mathcal{P}_x) \bullet (\mathcal{F}_y, \mathcal{P}_y) = \mathcal{M}_\emptyset \Rightarrow (\mathcal{F}_x, \mathcal{P}_x) = \mathcal{M}_\emptyset \wedge (\mathcal{F}_y, \mathcal{P}_y) = \mathcal{M}_\emptyset$

$\Rightarrow \mathcal{F}_x \cup \mathcal{F}_y = \emptyset \wedge \{p \cup q \mid p \in \mathcal{P}_x,\, q \in \mathcal{P}_y,\, p \cap \mathcal{F}_y = q \cap \mathcal{F}_x\} = \{\emptyset\}$
$\Rightarrow \mathcal{F}_x = \emptyset \wedge \mathcal{F}_y = \emptyset \wedge \{p \cup q \mid p \in \mathcal{P}_x,\, q \in \mathcal{P}_y\} = \{\emptyset\}$
$\Rightarrow (\mathcal{F}_x, \mathcal{P}_x) = \mathcal{M}_\emptyset \wedge (\mathcal{F}_y, \mathcal{P}_y) = \mathcal{M}_\emptyset$

- Non Cancellative: $\mathcal{M}_\emptyset \bullet (\emptyset, \emptyset) = (\emptyset, \emptyset) = (\emptyset, \emptyset) \bullet (\emptyset, \emptyset)$
- Idempotent: $(\mathcal{F}_x, \mathcal{P}_x) \bullet (\mathcal{F}_x, \mathcal{P}_x)$

$= (\mathcal{F}_x \cup \mathcal{F}_x, \{p \cup q \mid p \in \mathcal{P}_x,\, q \in \mathcal{P}_x,\, p \cap \mathcal{F}_x = q \cap \mathcal{F}_x\})$
$= (\mathcal{F}_x, \{p \cup q \mid p \in \mathcal{P}_x,\, q \in \mathcal{P}_x,\, p = q\})$
$= (\mathcal{F}_x, \mathcal{P}_x)$

- Zero element: $(X, \emptyset) \bullet (\mathcal{F}, \mathcal{P}) = (\mathcal{F}, \mathcal{P}) \bullet (X, \emptyset) = (X, \emptyset)$

Part 2: $\leq$ is a partial order on $\mathfrak{E}(X)$. For any $\mathcal{M}_x \leq \mathcal{M}_y \leq \mathcal{M}_z \in \mathfrak{E}(X)$, we have

- Reflexivity: $\mathcal{M}_x \bullet \mathcal{M}_\emptyset = \mathcal{M}_x$
- Antisymmetry: suppose additionally $\mathcal{M}_y \leq \mathcal{M}_x$. We have

$$\begin{aligned}
\mathcal{M}_x &= \mathcal{M}_y \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X)\\
&= \mathcal{M}_x \bullet \mathcal{M}' \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X)\\
&= \mathcal{M}_x \bullet \mathcal{M}' \bullet \mathcal{M}' \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X)\\
&= \mathcal{M}_y \bullet \mathcal{M}' \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X)\\
&= \mathcal{M}_y \bullet \mathcal{M} \bullet \mathcal{M}' && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X)\\
&= \mathcal{M}_x \bullet \mathcal{M}' && \text{for some } \mathcal{M}' \in \mathfrak{E}(X)\\
&= \mathcal{M}_y
\end{aligned}$$

- Transitivity:

$$\begin{aligned}
\mathcal{M}_z &= \mathcal{M}_y \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X)\\
&= (\mathcal{M}_x \bullet \mathcal{M}') \bullet \mathcal{M} && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X)\\
&= \mathcal{M}_x \bullet (\mathcal{M}' \bullet \mathcal{M}) && \text{for some } \mathcal{M}, \mathcal{M}' \in \mathfrak{E}(X)
\end{aligned}$$

Part 3: $(\mathfrak{E}(X), \leq)$ is a lattice with $\mathcal{M}_\emptyset$ as lower bound. As for all $\mathcal{M} \in \mathfrak{E}(X)$, we have $\mathcal{M} \bullet \mathcal{M}_\emptyset = \mathcal{M}$ which implies by definition that $\mathcal{M}_\emptyset \leq \mathcal{M}$. Therefore, $\mathcal{M}_\emptyset$ is the bottom. Surprisingly, the top is the zero FM: $(X, \emptyset)$ which the absorbent element !!!

- By definition, we have $\mathcal{M}_x \bullet \mathcal{M}_y \in \sup(\mathcal{M}_x) \cap \sup(\mathcal{M}_y)$. Moreover, with $\mathcal{M} \in \sup(\mathcal{M}_x) \cap \sup(\mathcal{M}_y)$, we have

$$\begin{aligned}
\mathcal{M} &= \mathcal{M}_x \bullet \mathcal{M}\\
&= \mathcal{M}_x \bullet (\mathcal{M}_y \bullet \mathcal{M})\\
&= (\mathcal{M}_x \bullet \mathcal{M}_y) \bullet \mathcal{M}
\end{aligned}$$

  And so we have $\mathcal{M}_x \sqcap \mathcal{M}_y = \mathcal{M}_x \bullet \mathcal{M}_y$.
  LP: $\mathcal{M}_x$, $\mathcal{M}_y \leq \mathcal{M}_x \bullet \mathcal{M}_y$ quindi parliamo del LUB (SUP, JOIN): giusto?
- Let $\mathcal{M} = (\mathcal{F}, \mathcal{P}) = (\mathcal{F}_x \cap \mathcal{F}_y, \{p_x \cap \mathcal{F}_y \mid p_x \in \mathcal{P}_x\} \cup \{p_y \cap \mathcal{F}_x \mid p_y \in \mathcal{P}_y\})$. We have $\mathcal{F} \subseteq \mathcal{F}_i$ and $\{p \cap \mathcal{F} \mid p \in \mathcal{P}_i\} \subseteq \mathcal{P}$ for $i \in \{x, y\}$: we thus have $\mathcal{M} \in \inf(\mathcal{M}_x) \cap \inf(\mathcal{M}_y)$. Moreover, with $(\mathcal{F}', \mathcal{P}') \in \inf(\mathcal{M}_x) \cap \inf(\mathcal{M}_y)$, we have that $\mathcal{F}' \subseteq \mathcal{F}_i$ and $\{p \cap \mathcal{F}' \mid p \in \mathcal{P}_i\} \subseteq \mathcal{P}'$ for $i \in \{x, y\}$. Hence, we have $\mathcal{F}' \subseteq \mathcal{F}$ and $\{p \cap \mathcal{F}' \mid p \in \mathcal{P}\} \subseteq \mathcal{P}'$, which implies that $(\mathcal{F}', \mathcal{P}') \leq \mathcal{M}$. And so we have $\mathcal{M}_x \sqcup \mathcal{M}_y = \mathcal{M}$. LP: $\mathcal{M} \leq \mathcal{M}_x$, $\mathcal{M}_y$! This a glb! Swap?
  LP: The proof is unclear to me, but it seems reasonable. It seems to rests on the fact that:
  LP: $(\phi_1 \to \phi) \wedge (\phi_2 \to \phi)$ is logically eqivalent to $(\phi_1 \vee \phi_2) \to \phi$

  The next theorem is the core of Theorem 16 .

Theorem 12. *For any set X and every $\mathcal{M}_x, \mathcal{M}_y \in \mathfrak{E}(X)$, with $\mathcal{M}_x = (\mathcal{F}_x, \mathcal{P}_x)$ and $\mathcal{M}_y = (\mathcal{F}_y, \mathcal{P}_y)$, the following properties are equivalent:*

i) $\mathcal{M}_x \leq \mathcal{M}_y$
ii) $\mathcal{M}_x \bullet \mathcal{M}_y = \mathcal{M}_y$
iii) $\mathcal{F}_x \subseteq \mathcal{F}_y \wedge \mathcal{P}_x \supseteq \{q \cap \mathcal{F}_x \mid q \in \mathcal{P}_y\}$

Proof. $i) \Rightarrow ii)$**.** We have

$$\begin{aligned}
\mathcal{M}_x \bullet \mathcal{M}_y &= \mathcal{M}_x \bullet (\mathcal{M}_x \bullet \mathcal{M}) && \text{for some } \mathcal{M} \in \mathfrak{E}(X) \\
&= (\mathcal{M}_x \bullet \mathcal{M}_x) \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X) \\
&= \mathcal{M}_x \bullet \mathcal{M} && \text{for some } \mathcal{M} \in \mathfrak{E}(X) \\
&= \mathcal{M}_y
\end{aligned}$$

$ii) \Rightarrow iii)$**.** By definition of $\bullet$, it is clear from the hypothesis that $\mathcal{F}_x \subseteq \mathcal{F}_y$. Moreover, if we write $S = \{q \cap \mathcal{F}_x \mid q \in \mathcal{P}_y\}$, the hypothesis give us that $\mathcal{P}_y = \{q \mid q \in \mathcal{P}_y, q \cap \mathcal{F}_x \in (\mathcal{P}_x \cap S)\}$. We can thus conclude with the following equivalences:

$$\begin{aligned}
&\mathcal{P}_y = \{q \mid q \in \mathcal{P}_y, q \cap \mathcal{F}_x \in (\mathcal{P}_x \cap S)\} \\
&\quad\Leftrightarrow \forall q \in \mathcal{P}_y,\, q \cap \mathcal{F}_x \in (\mathcal{P}_x \cap S) \\
&\quad\Leftrightarrow \{q \cap \mathcal{F}_x \mid q \in \mathcal{P}_y\} \subseteq (\mathcal{P}_x \cap S) \\
&\quad\Leftrightarrow S \subseteq (\mathcal{P}_x \cap S) \\
&\quad\Leftrightarrow S \subseteq \mathcal{P}_x
\end{aligned}$$

$iii) \Rightarrow i)$**.** Let still write $S = \{q \cap \mathcal{F}_x \mid q \in \mathcal{P}_y\}$. We have:

$$\begin{aligned}
\mathcal{M}_x \bullet \mathcal{M}_y &= (\mathcal{F}_x \cup \mathcal{F}_y, \{p \cup q \mid p \in \mathcal{P}_x,\, q \in \mathcal{P}_y,\, p \cap \mathcal{F}_y = q \cap \mathcal{F}_x\}) \\
&= (\mathcal{F}_y, \{q \mid q \in \mathcal{P}_y, q \cap \mathcal{F}_x \in \mathcal{P}_x\}) \\
&= (\mathcal{F}_y, \{q \mid q \in \mathcal{P}_y, q \cap \mathcal{F}_x \in (\mathcal{P}_x \cap S)\} \\
&\qquad \cup \{q \mid q \in \mathcal{P}_y,\, q \cap \mathcal{F}_x \in \{p \mid p \in \mathcal{P}_x, \forall q \in \mathcal{P}_y, p \neq q \cap \mathcal{F}_x\}\}) \\
&= (\mathcal{F}_y, \{q \mid q \in \mathcal{P}_y, q \cap \mathcal{F}_x \in (\mathcal{P}_x \cap S)\}) \\
&= (\mathcal{F}_y, \{q \mid q \in \mathcal{P}_y, q \cap \mathcal{F}_x \in S)\}) \\
&= \mathcal{M}_y
\end{aligned}$$

This implies, by definition of $\leq$, that $\mathcal{M}_x \leq \mathcal{M}_y$.

B — — — OTHER STUFF BY MICHAEL — — —

Theorem 13. *Given two sets X and Y with $Y \subseteq X$, the function Π_Y is: idempotent; monotonic increasing with respect to $\leq$ such that $\Pi_Y(\mathcal{M}) = \max_{\leq}(\inf(\mathcal{M}) \cap \mathfrak{E}(Y))$; and a morphism with respect to $\bullet$ iff $\#X \leq 1$.*

Proof. Let consider any feature model $\mathcal{M} = (\mathcal{F}, \mathcal{P}) \in \mathfrak{E}(X)$.

• Idempotent:

$$\begin{aligned}\Pi_Y(\Pi_Y(\mathcal{M})) &= ((\mathcal{F} \cap Y) \cap Y, \{(p \cap Y) \cap Y \mid p \in \mathcal{P}\}) \\ &= (\mathcal{F} \cap Y, \{p \cap Y \mid p \in \mathcal{P}\}) \\ &= \Pi_Y((\mathcal{F}, \mathcal{P})) = \Pi_Y(\mathcal{M})\end{aligned}$$

• $\Pi_Y(\mathcal{M}) \leq \mathcal{M}$:

$$\begin{aligned}\Pi_Y(\mathcal{M}) \bullet \mathcal{M} &= ((\mathcal{F} \cap Y) \cup \mathcal{F}, \{(p \cap Y) \cup q \mid p, q \in \mathcal{P} \wedge p \cap Y = q \cap Y\}) \\ &= (\mathcal{F}, \{q \mid p, q \in \mathcal{P} \wedge p \cap Y = q \cap Y\}) \\ &= (\mathcal{F}, \mathcal{P}) = \mathcal{M}\end{aligned}$$

• $\Pi_Y(\mathcal{M}) = \max_{\leq}(\inf(\mathcal{M}) \cap \mathfrak{E}(Y))$: consider $(\mathcal{F}', \mathcal{P}') \in \inf(\mathcal{M}) \cap \mathfrak{E}(Y)$, we have

$$\begin{cases}\mathcal{F}' \subset Y \wedge \mathcal{F}' \subset \mathcal{F} \Rightarrow \mathcal{F}' \subset Y \cap \mathcal{F} \\ \mathcal{P}' \supseteq \{p \cap \mathcal{F}' \mid p \in \mathcal{P}\} = \{(p \cap Y) \cap \mathcal{F}' \mid p \in \mathcal{P}\}\end{cases}$$

$\Rightarrow (\mathcal{F}', \mathcal{P}') \leq \Pi_Y(\mathcal{M})$

• Π_Y is a morphism when $\#X \leq 1$

$$\begin{cases}\mathfrak{E}(\emptyset) = \{(\emptyset, \emptyset), (\emptyset, \{\emptyset\})\} \\ \mathfrak{E}(\{a\}) = \mathfrak{E}(\emptyset) \cup \{(\{a\}, \emptyset), (\{a\}, \{\emptyset\}), (\{a\}, \{a\}),\}\end{cases}$$

It is easy to check that for any $\mathcal{M}_1, \mathcal{M}_2 \in \mathfrak{E}(X)$ and any $Y \subseteq X$, we have $\Pi_Y(\mathcal{M}_1 \bullet \mathcal{M}_2) = \Pi_Y(\mathcal{M}_1) \bullet \Pi_Y(\mathcal{M}_2)$

• Π_Y is not a morphism when $\#X > 1$. Consider $a \in X$, $Y = X \setminus \{a\}$, $\mathcal{M}_1 = (X, \{X\})$ and $\mathcal{M}_2 = (X, \{X \setminus \{a\}\})$. We have:

$$\begin{cases}\Pi_Y(\mathcal{M}_1 \bullet \mathcal{M}_2) = \Pi_Y((X, \emptyset)) = (Y, \emptyset) \\ \Pi_Y(\mathcal{M}_1) \bullet \Pi_Y(\mathcal{M}_2) = (Y, \{Y\}) \bullet (Y, \{Y\}) = (Y, \{Y\})\end{cases}$$

C — — — Luca Reasonings — — —

C.1 Semantic of composition in propositional FM

In accordance with redundant Definitions 2 and 12, we denote $\mathfrak{E}(X)$ the set of all *extensional* feature model $(\mathcal{F}, \mathcal{P})$ such that $\mathcal{F} \subseteq X$. In addition, we denote $\mathfrak{P}(X)$ the set of all *propositional* feature model $(\mathcal{F}, \phi)$ such that $\mathcal{F} \subseteq X$.

We remind that an interpretation (a.k.a. truth assignment or valuation) is a function which maps propositional logic variables to true or false.

Definition 13. *Let $(\mathcal{F}, \mathcal{P})$ be an extensional feature model. If $p \in \mathcal{P}$ then we denote $\mathcal{I}_p^{\mathcal{F}}$ the interpretation mapping $\mathcal{F}$ in true or false such that:*

$$\mathcal{I}_p^{\mathcal{F}}(f) = \begin{cases} true & if\ f \in p, \\ false & iff \in \mathcal{F} \setminus p.\end{cases}$$

Definition 14. *Let $\mathcal{F}_1, \mathcal{F}_2$ be sets of feature models and $p_1 \subseteq \mathcal{F}_1$ and $p_2 \subseteq \mathcal{F}_2$.*

- *If $\mathcal{F}_1 \subseteq \mathcal{F}_2$ then $\mathcal{I}_p^{\mathcal{F}_1} \subseteq \mathcal{I}_{p_2}^{\mathcal{F}_2}$ means that $\mathcal{I}_{p_1}^{\mathcal{F}_1}(f) = \mathcal{I}_{p_2}^{\mathcal{F}_2}(f)$, for all $f \in \mathcal{P}_1$.*
- *We say that $\mathcal{I}_{p_1}^{\mathcal{F}_1}, \mathcal{I}_{p_2}^{\mathcal{F}_2}(f)$ are compatible whenever $\mathcal{I}_{p_1}^{\mathcal{F}_1}(f) = \mathcal{I}_{p_2}^{\mathcal{F}_2}(f)$, for all $f \in \mathcal{P}_1 \cap \mathcal{P}_2$.*

Definition 15. *Let $(\mathcal{F}, \phi) \in \mathfrak{P}(X)$. We denote $\texttt{ext}(\mathcal{F}, \phi)$ denotes its corresponding extensional feature model $(\mathcal{F}, \mathcal{P})$, where $\mathcal{P}$ is the set of all and only products p such that $\mathcal{I}_p^{\mathcal{F}} \models \phi$.*

We write $\phi_1 \equiv \phi_2$ whenever ϕ_1, ϕ_2 are logically equivalent. Note that, if $(\mathcal{F}_1, \phi_1), (\mathcal{F}_2, \phi_2) \in \mathfrak{P}(X)$ then: $\mathtt{ext}(\mathcal{F}, \phi_1) = \mathtt{ext}(\mathcal{F}, \phi_2)$ iff $\phi_1 \equiv \phi_2$.

Formalizing the implicit characterization of Example 3.

This characterization is not presented anywhere: SBAGLIO?

Theorem 14 (Semantic of composition). *If* $(\mathcal{F}_1, \phi_1), (\mathcal{F}_2, \phi_2) \in \mathfrak{P}(X)$ *then*

$$\mathtt{ext}(\mathcal{F}_1, \phi_1) \bullet \mathtt{ext}(\mathcal{F}_2, \phi_2) = \mathtt{ext}(\mathcal{F}_1 \cup \mathcal{F}_2, \phi_1 \wedge \phi_2).$$

Proof. Let $\mathtt{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$, for $i = 1, 2$. By Definition 12,

$$(\mathcal{F}_1, \mathcal{P}_1) \bullet (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_1 \cup \mathcal{F}_2, \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 \cap \mathcal{F}_2 = p_2 \cap \mathcal{F}_1\}).$$

But $p_1 \cap \mathcal{F}_2 = p_2 \cap \mathcal{F}_1$ iff $\mathcal{I}_{p_1}^{\mathcal{F}_2}, \mathcal{I}_{p_2}^{\mathcal{F}_2}$ are compatible. Therefore $\mathcal{I}_{p_1}^{\mathcal{F}_2}, \mathcal{I}_{p_2}^{\mathcal{F}_2}$ can be extended in a common interpretation $\mathcal{I}_{p_1 \cup p_2}^{\mathcal{F}_1 \cap \mathcal{F}_2}$, namely $\mathcal{I}_{p_1}^{\mathcal{F}_2}, \mathcal{I}_{p_2}^{\mathcal{F}_2} \subseteq \mathcal{I}_{p_1 \cup p_2}^{\mathcal{F}_1 \cap \mathcal{F}_2}$. Clearly, $\mathcal{I}_{p_1 \cup p_2}^{\mathcal{F}_1 \cap \mathcal{F}_2} \models \phi_i$ for $i = 1, 2$ and $\mathcal{I}_{p_1 \cup p_2}^{\mathcal{F}_1 \cap \mathcal{F}_2} \models \phi_1 \wedge \phi_2$ so the proof is done. □

C.2 New-order characterization

Let $\leq$ the partial order defined in sub-Section 2.4 on feature models. Just to remind: $\mathcal{M} \leq \mathcal{M}'$ iff $\exists \mathcal{M}''$ with $\mathcal{M}' = \mathcal{M} \bullet \mathcal{M}''$. By Theorem 12, we know that $\mathcal{M} \bullet \mathcal{M}' = \mathcal{M}'$.

The above relation is explained by the following lemma.

Theorem 15. *Let* $(\mathcal{F}_1, \phi_1), (\mathcal{F}_2, \phi_2) \in \mathfrak{P}(X)$ *and let* $\mathtt{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$, *for* $i = 1, 2$.

- *If* $\mathcal{F}_1 = \mathcal{F}_2$ *then the following statements are equivalent:*
  1. $(\mathcal{F}_1, \mathcal{P}_1) \leq (\mathcal{F}_2, \mathcal{P}_2)$;
  2. $\mathcal{P}_2 \subseteq \mathcal{P}_1$;
  3. $\models \phi_2 \rightarrow \phi_1$.
- *If* $\mathcal{F}_1 \subseteq \mathcal{F}_2$ *then the following statements are equivalent:*
  1. $(\mathcal{F}_1, \mathcal{P}_1) \leq (\mathcal{F}_2, \mathcal{P}_2)$;
  2. $\mathcal{P}_2|_{\mathcal{F}_1} \subseteq \mathcal{P}_1$;
  3. $\mathcal{I}_{p_2}^{\mathcal{F}_2} \models \phi_1$, *for all* $p_2 \in \mathcal{P}_2$;
  4. $\models \phi_2 \rightarrow \phi_1$;
  5. *if* $q \models \phi_2$ *then* $q \models \phi_1$.

Proof. – **First, we prove that** *1.* **iff** *2.* By Theorem 12, we know that

$$\begin{aligned}\mathcal{P}_2 &= \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 \cap \mathcal{F}_2 = p_2 \cap \mathcal{F}_1\} \\ &= \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 = p_2\} \\ &= \{p_2 \in \mathcal{P}_2 \mid p_1 \in \mathcal{P}_1, p_1 = p_2\} \qquad \subseteq \mathcal{P}_1\end{aligned}$$

Second, we prove that *2.* **iff** *3.* We know that $p_i \in \mathcal{P}_i$ iff $\mathcal{I}_{p_i}^{\mathcal{F}_i} \models \phi_i$. Moreover, $\models \phi_2 \rightarrow \phi_1$ means that $\mathcal{I}_{\mathcal{I}}^{p_1} \models \phi_1$ implies $\mathcal{I}_{\mathcal{I}}^{p_1} \models \phi_2$. Therefore $\models \phi_2 \rightarrow \phi_1$ exactly when $\mathcal{P}_2 \subseteq \mathcal{P}_1$.

– **First, we prove that** *1.* **iff** *2.* By Theorem 12, we know that

$$\begin{aligned}\mathcal{P}_2 &= \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 \cap \mathcal{F}_2 = p_2 \cap \mathcal{F}_1\} \\ &= \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 = p_2 \cap \mathcal{F}_1\} \\ &= \{p_2 \in \mathcal{P}_2 \mid p_1 \in \mathcal{P}_1, p_1 = p_2 \cap \mathcal{F}_1\}\end{aligned}$$

Second, we prove that *2.* **iff** *3.*
Similar to the analogous point before.

C.3 Schroter-order characterization

In accordance with the interface Schröter et al. [25]:
$(\mathcal{F}_1, \mathcal{P}_1) \preceq (\mathcal{F}_2, \mathcal{P}_2)$ iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\{p \cap \mathcal{F}_1 \mid p \in \mathcal{P}_2\}) = \mathcal{P}_1$.

Theorem 16. *Let $(\mathcal{F}_1, \phi_1), (\mathcal{F}_2, \phi_2) \in \mathfrak{P}(X)$ and let $\boldsymbol{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$, for $i = 1, 2$.*

- *If $\mathcal{F}_1 = \mathcal{F}_2$ then the following statements are equivalent:*
  1. *$(\mathcal{F}_1, \mathcal{P}_1) \preceq (\mathcal{F}_2, \mathcal{P}_2)$;*
  2. *$\mathcal{P}_2 = \mathcal{P}_1$;*
  3. *$\models \phi_2 \leftrightarrow \phi_1$ or, in other words, ϕ_1, ϕ_2 are logically equivalent (i.e. $\phi_1 \equiv \phi_2$);*
  4. *$(\mathcal{F}_2, \mathcal{P}_2) \preceq (\mathcal{F}_1, \mathcal{P}_1)$.*
- *If $\mathcal{F}_1 \subseteq \mathcal{F}_2$ then the following statements are equivalent:*
  1. *$(\mathcal{F}_1, \mathcal{P}_1) \preceq (\mathcal{F}_2, \mathcal{P}_2)$;*
  2. *$\mathcal{P}_2|_{\mathcal{F}_1} = \mathcal{P}_1$;*
  3. *$\models \phi_2 \to \phi_1$ and for all $p_1 \in \mathcal{P}_1$ there is a $p_2 \in \mathcal{P}_2$ such that $\mathcal{I}_{p_2}^{\mathcal{P}_2} \subseteq \mathcal{I}_{p_1}^{\mathcal{P}_1}$.*
  4. *$\models (\bigvee_{\mathcal{P}_2 \setminus \mathcal{P}_1} \phi_2) \leftrightarrow \phi_1$ where: $(\bigvee_{\emptyset} \varphi) = \varphi$ and $(\bigvee_{S} \varphi) = (\bigvee_{S - \{x\}} \varphi[x := true])$.*

Proof. TBD

C.4 Comparing the two orders

In accordance with the interface Schröter et al. [25], we defined in Definition **??** that: $\mathcal{M}_1 \preceq \mathcal{M}_2$, whenever there exists Y such that $\mathcal{M}_1 = (\mathcal{F} \cap Y, \{p \cap Y \mid p \in \mathcal{P}\})$.

Lemma 5. *Let $(\mathcal{F}_1, \mathcal{P}_1), (\mathcal{F}_2, \mathcal{P}_2) \in \mathfrak{E}(X)$.*
$(\mathcal{F}_1, \mathcal{P}_1) \preceq (\mathcal{F}_2, \mathcal{P}_2)$ iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\{p \cap \mathcal{F}_1 \mid p \in \mathcal{P}_2\}) = \mathcal{P}_1$.

Proof. Trivial, if the new definition is equivalent to the old one. □

But in accord with Theorem 12, we have:
$(\mathcal{F}_1, \mathcal{P}_1) \leq (\mathcal{F}_2, \mathcal{P}_2)$ iff $\mathcal{F}_1 \subseteq \mathcal{F}_2$ and $\{p \cap \mathcal{F}_1 \mid p \in \mathcal{P}_2\} \subseteq \mathcal{P}_1$

Theorem 17. *If $\mathcal{M}_1 \preceq \mathcal{M}_2$ implies $\mathcal{M}_1 \leq \mathcal{M}_2$.*

Proof. Immediate, by Lemma 5 and Theorem 12. □

C.5 Other possible compositions

The semantics above suggest some other possible composition of feature models. We wonder if there are operations $+, \odot$ on extensional feature models satisfying:

$$\begin{aligned}\mathtt{ext}(\mathcal{F}_1, \phi_1) \odot \mathtt{ext}(\mathcal{F}_2, \phi_2) = \mathtt{ext}(\mathcal{F}_1 \cup \mathcal{F}_2, \phi_1 \vee \phi_2),\\ \mathtt{ext}(\mathcal{F}_1, \phi_1) + \mathtt{ext}(\mathcal{F}_2, \phi_2) = \mathtt{ext}(\mathcal{F}_1 \cup \mathcal{F}_2, \phi_1 \oplus \phi_2).\end{aligned}$$

The disjunction. We denote $X \cdot Y$ the set $\{x \cup y \mid x \in X, y \in Y\}$. In a pure logic perspective, we can consider the set of interpretations satisfying the disjunction:

$$(\mathcal{F}_1, \mathcal{P}_1) \odot (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_1 \cup \mathcal{F}_2, \mathcal{P}_1 \cdot 2^{(\mathcal{F}_2 \setminus \mathcal{F}_1)} \cup \mathcal{F}_2 \cdot 2^{(\mathcal{F}_1 \setminus \mathcal{F}_2)})$$

However, in the SPL context we are (possibly) including set of features that projected (restricted to relevant $\mathcal{F}_i$) pick up product not considered in the original FMs.

C.6 Lattice-algebraic

Consider the next two operations:

$$\begin{aligned}(\mathcal{F}_1, \mathcal{P}_1) \bullet (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_1, \mathcal{P}_1) \sqcup (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_1 \cup \mathcal{F}_2, \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 \cap \mathcal{F}_2 = p_2 \cap \mathcal{F}_1\})\\ (\mathcal{F}_1, \mathcal{P}_1) \star (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_1, \mathcal{P}_1) \sqcap (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_1 \cap \mathcal{F}_2, \{p_1 \cap \mathcal{F}_2 \mid p_1 \in \mathcal{P}_1\} \cup \{p_2 \cap \mathcal{F}_1 \mid p_2 \in \mathcal{P}_2\})\end{aligned}$$

We can prove what follows.

- Commutativity of $\sqcup$ and $\sqcap$ are trivial.
- Associativity of $\sqcup$ is in [15, Appendix A, p.225]. Associativity of $\sqcup$ is quite evident.
- Idempotency of $\sqcup$ and $\sqcap$ are easy.
- Absorption (Sembrerebbe di si)

  $$\begin{aligned}&(\mathcal{F}_1, \mathcal{P}_1) \bullet ((\mathcal{F}_1, \mathcal{P}_1) \star (\mathcal{F}_2, \mathcal{P}_2))\\ &= (\mathcal{F}_1, \mathcal{P}_1) \bullet (\mathcal{F}_1 \cap \mathcal{F}_2, \{p_1 \cap \mathcal{F}_2 \mid p_1 \in \mathcal{P}_1\} \cup \{p_2 \cap \mathcal{F}_1 \mid p_2 \in \mathcal{P}_2\}) = (\mathcal{F}_1, \mathcal{P}_1)\end{aligned}$$

  $$\begin{aligned}&(\mathcal{F}_1, \mathcal{P}_1) \star ((\mathcal{F}_1, \mathcal{P}_1) \bullet (\mathcal{F}_2, \mathcal{P}_2))\\ &= (\mathcal{F}_1, \mathcal{P}_1) \star (\mathcal{F}_1 \cup \mathcal{F}_2, \{p_1 \cup p_2 \mid p_1 \in \mathcal{P}_1, p_2 \in \mathcal{P}_2, p_1 \cap \mathcal{F}_2 = p_2 \cap \mathcal{F}_1\}) = (\mathcal{F}_1, \mathcal{P}_1)\end{aligned}$$

Thus they form a lattice, and morever a bounded lattice:

- The top is $(X, \emptyset)$ because $(X, \emptyset) \star (\mathcal{F}, \mathcal{P}) = (\mathcal{F}, \mathcal{P})$ is immediate. Moreover, $(X, \emptyset) \bullet (\mathcal{F}, \mathcal{P}) = (X, \emptyset)$.
- The bottom $(\emptyset, \{\emptyset\})$ because $(\emptyset, \{\emptyset\}) \bullet (\mathcal{F}, \mathcal{P}) = (\mathcal{F}, \mathcal{P})$ is immediate. Moreover, $(\emptyset, \{\emptyset\}) \star (\mathcal{F}, \mathcal{P}) = (\emptyset, \{\emptyset\})$.

These operations do not form a distributive lattice, because they are not cancellative!

Definition 16 (Heyting Pseudocomplement). *If* $\mathcal{M}_1 = (\mathcal{F}_1, \mathcal{P}_1) \in \mathfrak{E}(X)$ *then the set* $\{ (\mathcal{F}_2, \mathcal{P}_2) \mid (\mathcal{F}_1, \mathcal{P}_1) \star (\mathcal{F}_2, \mathcal{P}_2) = (\emptyset, \{\emptyset\}) \}$ *is:*

- $\{(\mathcal{F}_2, \mathcal{P}_2) \mid \mathcal{F}_2 \subseteq X \setminus \mathcal{F}_1$ *and* $\mathcal{P}_2 \subseteq 2^{\mathcal{F}_2}\}$ *whenever* $\mathcal{P}_2 \neq \emptyset$;
- $\{(\mathcal{F}_2, \mathcal{P}_2 \cup \{\emptyset\}) \mid \mathcal{F}_2 \subseteq X \setminus \mathcal{F}_1$ *and* $\mathcal{P}_2 \subseteq 2^{\mathcal{F}_2}\}$ *whenever* $\mathcal{P}_2 = \emptyset$.

Since $\max\{ (\mathcal{F}_2, \mathcal{P}_2) \mid (\mathcal{F}_1, \mathcal{P}_1) \star (\mathcal{F}_2, \mathcal{P}_2) = (\emptyset, \{\emptyset\}) \}$ *is the pseudocomplement of* $\mathcal{M}$, *we have:* $\mathcal{M}_1^* = \{(X \setminus \mathcal{F}_1, \emptyset)\}$ *whenever* $\mathcal{P}_2 \neq \emptyset$; $\mathcal{M}_1^* = \{(X \setminus \mathcal{F}_1, \{\emptyset\})\}$ *whenever* $\mathcal{P}_2 = \emptyset$.

Proposition 1 (Corollary p.104 of [11]).

- *If* $\mathcal{P}_1 = \mathcal{P}_2 = \emptyset$ *then*

$$\begin{aligned}&(\mathcal{F}_1, \mathcal{P}_1) \sqcap ((\mathcal{F}_1, \mathcal{P}_1) \sqcap (\mathcal{F}_2, \mathcal{P}_2))^* = (\mathcal{F}_1, \mathcal{P}_1) \sqcap (\mathcal{F}_1 \cap \mathcal{F}_2, \emptyset)^* = \\ &(\mathcal{F}_1, \mathcal{P}_1) \sqcap (X \setminus \mathcal{F}_1 \cap \mathcal{F}_2, \{\emptyset\}) = (\mathcal{F}_1 \setminus \mathcal{F}_2, \{\emptyset\}) \\ &(\mathcal{F}_1, \mathcal{P}_1) \sqcap (X \setminus \mathcal{F}_2, \{\emptyset\}) = (\mathcal{F}_1, \mathcal{P}_1) \sqcap (\mathcal{F}_2, \mathcal{P}_2)^*\end{aligned}$$

  If $\mathcal{P}_1 \neq \emptyset$ *and* $\mathcal{P}_2 = \emptyset$ *then*
  If $\mathcal{P}_2 \neq \emptyset$ *and* $\mathcal{P}_1 = \emptyset$ *then*
  If $\mathcal{P}_1, \mathcal{P}_2 \neq \emptyset$ *then*

- $(\mathcal{F}_1, \mathcal{P}_1) \star (X, \emptyset) = (\mathcal{F}_1, \mathcal{P}_1)$
- $(\emptyset, \{\emptyset\})^* = (X, \emptyset)$ *and* $(X, \emptyset)^* = (\emptyset, \{\emptyset\})$.

C.7 Other Logical characterizations

Let $\mathcal{M} = (\mathcal{F}, \mathcal{P})$ be a feature model. The *slice* operator Π_Y on feature models, where Y is a set of features, is defined by: $\Pi_Y(\mathcal{M}) = (\mathcal{F} \cap Y, \mathcal{P} \mid_Y)$ where $\mathcal{P} \mid_Y = \{p \cap Y \mid p \in \mathcal{P}\}$.

Theorem 18. *Let* $(\mathcal{F}_i, \phi_i) \in \mathfrak{P}(X)$ *and let* $\texttt{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$ *for* $i = 1, 2$. *If* $\Pi_Y(\mathcal{F}_1, \mathcal{P}_1) = (\mathcal{F}_2, \mathcal{P}_2)$ *then* $\models (\bigvee_{\mathcal{F}_1 \setminus \mathcal{F}_2} \phi_1) \leftrightarrow \phi_2$.

Note that $(\mathcal{F}_1, \mathcal{P}_1) \star (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_1 \cap \mathcal{F}_2, \mathcal{P}_1 \mid_{\mathcal{F}_2} \cup \mathcal{P}_2 \mid_{\mathcal{F}_1})$, so the next corollary hold.

Corollary 2. *Let* $(\mathcal{F}_i, \phi_i) \in \mathfrak{P}(X)$ *and let* $\texttt{ext}(\mathcal{F}_i, \phi_i) = (\mathcal{F}_i, \mathcal{P}_i)$ *for* $i = 1, 2, 3$. *If* $(\mathcal{F}_1, \mathcal{P}_1) \star (\mathcal{F}_2, \mathcal{P}_2) = (\mathcal{F}_3, \mathcal{P}_3)$ *then* $\models \big((\bigvee_{\mathcal{F}_1 \cap \mathcal{F}_2} \phi_1) \leftrightarrow \phi_3\big) \vee \big((\bigvee_{\mathcal{F}_1 \cap \mathcal{F}_2} \phi_2) \leftrightarrow \phi_3\big)$

Theorem 19. *Let* $(\mathcal{F}, \phi) \in \mathfrak{P}(X)$.

- *If* $\texttt{ext}(\mathcal{F}, \phi) = (\emptyset, \{\emptyset\})$ *then* $\phi \equiv$ *true*.
- *If* $\texttt{ext}(\mathcal{F}, \phi) = (X, \{\})$ *then* $\phi \equiv$ *false*.

C.8 We do not have a ring!

- $\bullet$ is the MULTIPLICATION and $\mathcal{M}_\emptyset = (\emptyset, \{\emptyset\})$ is the ONE
- $\star$ is the SUM and $\mathcal{M}^X = (X, \emptyset)$ is the ZERO

Let $\mathcal{M} = (\mathcal{F}, \mathcal{P})$. We wonder if it has opposite: it clear that $(\mathcal{F}, \mathcal{P}) \star (-\mathcal{F}, -\mathcal{P}) = (X, \emptyset)$ in general ha no solution, because $\mathcal{F} \cap -\mathcal{F}$ is sporadically equal to X. This implies that we do not have a boolean algebra.